CRANFIELD UNIVERSITY

Aurelio Fernandes

Time Multigrid Methods and D-Adaptivity for Coupled Fluid Flow Solvers

SCHOOL OF MECHANICAL ENGINEERING

PhD Thesis

CRANFIELD UNIVERSITY
SCHOOL OF MECHANICAL ENGINEERING
Applied Mathematics and Computing Group

PhD Thesis

Academic Year 2000-2001

Aurelio Fernandes

Time Multigrid Methods and D-Adaptivity for Coupled Fluid Flow Solvers

Supervisor: Professor C. P. Thompson

February 2002

This thesis is submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

Abstract

This Thesis is about the application of coupled multigrid solvers to the numerical simulation of viscous incompressible fluids. In the centre of discussion is the adaptivity between a one-dimensional solver and a two-dimensional one.

The methodology used has proved highly successful for single-and multi-phase laminar flows, leading to solution algorithms that are robust, efficient and accurate. The solvers presented here required a considerable number of algorithmic developments. Some of them have demanded the use of some well-known software packages.

The Thesis outline is as follows: firstly, the modelling of transient single-phase and multi-phase flows is reviewed, together with a brief overview of the numerical schemes and multigrid methods used in the solvers. Secondly, the Navier-Stokes governing equations are presented and the space discretization formulas based on a control volume are formulated. After having specified the solution algorithms we present results for each solver for a set of test cases of varying complexity. Comparison with our reference commercial code is outlined, showing good agreement in the results.

Interpolation transfer operators used in the interface between the one-dimensional solver and the two-dimensional one are addressed. The coupled solvers are then applied on the numerical simulation of the transient flows on two complex multi-domain problems. Comparison results with the two-dimensional solvers have been performed.

The question of performance and accuracy is addressed in detail, both in terms of robustness and speed of convergence. Good accelerations are obtained using the coupled solver. The CPU-time spent to reach the expected steady-state solution is about ten to thirty five percent of the equivalent two-dimensional solver.

Considerable gains in memory usage have been achieved. The robustness has been easily verified in the comparison process with the two-dimensional transient solvers. Analytic solutions have been formulated and discussed.

However some dependence on the Reynolds numbers has been observed. This was due to the geometric constraints of the complex test cases and the change of some fluid properties.

Acknowledgements

First of all, I would like to thank particularly my supervisor, Professor C. P. Thompson both for the guidance and encouragement I received during this project. Additional thanks to the AMAC group which have been an important support for some critical tasks. Particularly important has been the contribution of Dr Tim Myers in the formulation of the asymptotic analysis. Many thanks to Rice University/Argonne National Laboratory for the use of the automatic differentiation package `Adifor`, which has been important for the prosecution of this thesis.

Naturally I am also indebted to Fundacao Fernando Pessoa and Prodep which financially contributed to the overall budget of this project. Further thanks to Group BCP, which gave me a leave to conclude the computational work and write this thesis.

To my family, particularly my wife Cecilia and my sons Hugo and Daniel I am grateful for their patience while I was several times absent in their lives. Finally, due to the enormous financial effort that I had to make, I must thank to my parents, which gave me a high education and enormous help, particularly in the beginning of my working life.

Contents

List of Figures

List of Tables

Notation

$\mathbf{v}$	Velocity vector
u	Horizontal component of the velocity vector
v	Vertical component of the velocity vector
p	Pressure
r	Volume Fraction
ρ	Density
μ	Viscosity
ν	Kinematic viscosity
Ω^h	Grid identifier
∇	Gradient operator
$\nabla.$	Divergence operator
$\mathbf{u}$	Generic vector
$\mathbf{T}$	Stress tensor
f	Body force density
σ	Surface tension
$\mathbf{n}$	Unit normal
Γ_α	Inter-phase mass transfer
$\mathbf{M}_\alpha$	Inter-phase momentum tranfer
$\tilde{\tau_\alpha}$	Stress tensor
$\mathbf{I}$	Identity tensor
$\mathbf{M}_{int,\alpha}$	Interfacial force density
$\mathbf{T}_\alpha$	Laminar shear stress tensors
$\mathbf{T}_\alpha^t$	Turbulent shear stress tensors
g	gravity acceleration
T^w	Wall friction
λ_k	Eigenvalue of a matrix
M	Number of phases
t	time
Δp	Pressure correction
Δx	Horizontal mesh size
Δy	Vertical mesh size
Δt	Time step
$\mathbf{e}^n$	Error vector of iteration n
$\rho(S)$	Spectral radius of a matrix S
Ω^h	Grid Identifier

$\mathcal{L}$	Continuous differential operator
$\mathcal{L}^k$	Discrete operator
I_{k-1}^k	Prolongation operator for the solution
$\hat{I}_k^{k-1}$	Restriction operator for the solution
I_k^{k-1}	Restriction operator for the residual
u^k	Solution vector
r^k	Residual vectot
$\tilde{\mathbf{u}}^k$	Approximation vector
$\delta\tilde{\mathbf{u}}^k$	Correction vector
τ_k^{k-1}	Defect on the level $k-1$ grid
τ^{k-1}	Truncation error on the level k grid
$\mathbf{L}^n$	Non-linear operator at time step n
$\mathbf{U}^n$	Vector of the components on a cell (i,j) at time step n
$\tilde{\mathbf{U}}^n$	Vector of the approximations of the components on a cell (i,j) at time step n
$C_{\alpha\beta}$	Interphase momentum transfer coefficient
C_D	Drag coefficient
$d_{\alpha\beta}$	Interfacial length
ω	Relaxation parameter
ϕ	Solution vector
$\phi^{(0)}$	Initial approximation of the solution
$\Delta\phi$	Newton correction
J	Jacobian matrix
$Nvis$	Number of level k quadrants unrefined
γ	Refinement parameter
λ	Scaling factor
λ_r	Relaxation parameter
I	Identity matrix
$\mathbf{u}_k^n$	Unknown in grid k for time step n
$\tilde{\mathbf{u}}_k{}^n$	Approximation of the unknown $\mathbf{u}_k^n$
$\mathbf{r}_k$	Residual in grid k
$R(\tilde{\mathbf{u}}^k)$	Column vector with the residuals of the approximations of the unknowns
$\|\ .\ \|_2$	Euclidean norm
$\|.\|$	Vector norm
$<>$	Averaging process
$\tilde{.}$	Phasic Averaged quantity
$\hat{.}$	Mass-weighted Averaged quantity
Re	Reynolds Number
$\overline{u(x)}$	One-dimensional Velocity
$\overline{p(x)}$	One-dimensional Pressure
U	Velocity scaling
P	Pressure scaling
H	Vertical scaling (y)
s	Volume Fraction (in cylindrical expressions)
w	Azimuthal velocity component (in cylindrical expressions)
R	Radius

Subscripts

α	Phase identifier(α)
β	Phase identifier(β)
int	Interfacial quantity
i	Horizontal index for a two-dimensional vector
j	Vertical index for a two-dimensional vector
t	Partial derivative with respect to the variable t
x	Partial deivative with respect to the variable x
y	Partial deivative with respect to the variable y
$itf1$	Interface line between one-dimensional region/two-dimensional region
$itf2$	Interface line between two-dimensional region/one-dimensional region

Superscripts

$*$	Approximate value
(cor)	Corrected value
n	Iteration number
h	Mesh size
T	Transpose operation
(c)	Coarse grid value
(f)	Fine grid value

Definitions

Forward differences

$$\Delta_{+t}u(x,t) = u(x,t+\Delta t) - u(x,t),$$

$$\Delta_{+x}u(x,t) = u(x+\Delta x,t) - u(x,t).$$

Backward differences

$$\Delta_{-t}u(x,t) = u(x,t) - u(x,t-\Delta t),$$

$$\Delta_{-x}u(x,t) = u(x,t) - u(x-\Delta x,t).$$

Central differences

$$\delta_t u(x,t) = u(x,t+\frac{1}{2}\Delta t) - u(x,t-\frac{1}{2}\Delta t),$$

$$\delta_x u(x,t) = u(x+\frac{1}{2}\Delta x,t) - u(x-\frac{1}{2}\Delta x,t).$$

Vector product

$$\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^T$$

Chapter 1

Introduction

The understanding of the physics of fluid flow is crucial for many applications in science and engineering. Several models of varying complexity have been developed. The process of choosing a model depends mainly in the particular problem under study. The phenomena of interest are described by governing equations, which are partial differential equations. The solution of such equations is very difficult since it requires considerable experience at least in three distinct disciplines: physics[1, 2], mathematics[3, 4] and numerical analysis[5, 6]. This fact originated the development of the Computational Fluid Dynamics field (see, for instance, [7, 8, 9]). Many fundamental issues have been solved. However, progress is ongoing. Although we have nowadays more computing power available, the question of performance can not be neglected. This is particular important when we compare solution algorithms for solving a problem.

A possible approach for solving problems of complex geometries is to decompose the entire domain in several domains, which can be treated by different solvers. This implies that transfer mechanisms between the solvers must be implemented, ensuring the consistency of the solution over the entire domain. The key idea can be to divide the entire domain of the problem in a way that we can treat each region of the problem with different dimensional solvers. This is the basis of what we call D-adaptivity for coupled fluid flow solvers. An example where such methodology can be thought of is the simulation of fluid flow through a contraction. Experiments suggest that in this kind of problem we can decompose the overall domain in one domain where the fluid flow can be treated by a one-dimensional solver, another domain where a two-dimensional solver must be used and finally a region where it is possible to come back to the one-dimensional solver. Naturally, it is expected that the time taken to reach the solution by this `D-adaptive solver` be less than the time spent by a two-dimensional solver when solving the same problem. However, the cost is the less accurate solution for the problem. This is due to the interpolation errors that occur when transferring the relevant information between the solvers. The relevance of a `D-adaptive solver` is even more evident when the overall domain of the problem is very large. Furthermore, the simulation of fluid flow in very long pipes is impossible to be done by a single two or three-dimensional solver. So the decomposition of the domain has to be taken. Another example

where the D-adaptivity can be implemented is the simulation of fluid flow through a T-junction with two inlets. In this problem it is still possible to decompose the overall domain in several domains in order to apply different dimensional solvers to each one.

This thesis covers the simulation of transient laminar incompressible single-and multi-phase flows in pipes. In the first case, a single substance is flowing. This may be a pure element or a homogeneous mixture such as air. In the second case, several fluids or conceptually similar materials are flowing at the same time. The main objective of this project was to create a `D-adaptive solver` that could be used in the simulation of such flows. The key idea was to couple a one-dimensional solver with a two-dimensional one. The main issue was to establish a set of rules that could be used in the dimensional changing interface. Obtaining a good coupled solver is not a trivial task. It should be[10]:

Accurate the discrete solution is as close to the solution of the continuous equations, as computational cost will allow.

Efficient the computational cost of the solution is proportional to the amount of physical change in the solution[11].

Robust it successfully simulates a wide range of flows without relying on user expertise and/or experimentation because the domain of convergence is good.

Self-adaptive the information on the solution obtained as the algorithm proceeds can be used to direct the computational effort without external intervention from the user.

For steady flows one such solver can be constructed using quasi-Newton coupled solvers associated with multigrid methods. This class of methods has been developed in recent years to correct a fundamental weakness of traditional solution algorithms for fluid flows: the slowing of convergence due to the iteration of each equation independently from the others. Newton solvers are a possible solution to this problem. The global system of non-linear equations is solved by Newton's method so that the coupling between the variables can take place. However, large linear systems are generated so that the computational cost can be heavy unless the sparseness of the Jacobian matrices can be exploited. Even if the sparseness is exploited to the full, these methods require a large amount of computing power. This variant is sometimes referred to as full quasi-Newton. However in local quasi-Newton solvers, the global non-linear system is solved cell by cell so that only local Jacobians are needed. So the correction systems are very small and can be solved using direct methods such as Gaussian elimination. Local quasi-Newton methods are often combined with multigrid methods, which provide significant acceleration. So multigrid complements the quasi-Newton solver by quickly resolving the long wavelengths of the error. Quasi-Newton coupled multigrid solvers are optimal order-wise, meaning that their complexity is of the order of the number of

unknowns. Hence if N is the number of unknowns of the discrete equations their complexity is $\mathcal{O}(N)$.

The variant of multigrid implemented is the Full Approximation Storage method. It has two advantages: firstly, it is directly applicable to non-linear problems and secondly, it automatically provides estimates for the truncation error of the solution. With this information grid refinement can be implemented where appropriate. So, error control can be achieved automatically. Moreover, Full Approximation Storage and more generally multi-level methods provide an environment for adaptive computations, since adaptation can be easily implemented by prolongation on sub-domains.

The methodology outlined above have been implemented in the computer codes `Pamg` and `Pamg-multiphase` which are described in [12, 13, 14, 15] and [10]. These codes solve two-dimensional Navier-Stokes equations for single- and multi-phase viscous incompressible fluids for Cartesian geometries.

Given this general perspective, one of our first goals was to develop two transient solvers to simulate single-and multi-phase flows. `Pamg` and `Pamg-multiphase` have been the starting codes leading to the correspondent transient solvers. New issues have emerged:

- The system of equations to solve is much more non-linear due to the existence of the time dependent unknowns. This originate additional terms in the coefficients of the system of equations to solve.

- The convergence of the solution to the steady case solution originates singular Jacobian matrixes which have to be handled.

- The increase on the amount of CPU-time spent to solve the systems is an additional factor to keep in mind when choosing the test-cases.

Having modified these two solvers to include an implicit scheme on time, the investigation on the robustness, efficiency and accuracy has been addressed with the simulation of single-and multi-phase flows on complex geometries.

The next step, has been the building architecture plan of the `D-adaptive solver`. The one-dimensional solver chosen has been the Eulerian Multiphase Adaptive Pipeline Solver `Emaps`. A complete description of this solver can be found in references [16, 17]. The final goal was to develop and validate the `D-adaptive solver`. The main issues to deal with are:

- The establishment of the correct interpolation formulas when changing from the one-dimensional to the two-dimensional environment, and vice-versa.

- The decision process of creating the dimensional interface.

- The correct validation process so that it can be classified as a good solver in the definition outlined above.

In Chapter Two the physical modelling and the numerical simulation of transient single-and multi-phase flows are examined. The mathematical basis for the different numerical techniques implemented in the transient solvers are presented. An incursion on the multigrid methods used is also outlined. The known kinds of adaptivity are briefly referenced. This literature review does not aim to be exhaustive. In fact, it highlights the most important issues in the simulation of transient fluid flows and explains why the `transient Pamg` codes and `D-adaptive solver` perform well as solvers for fluid flows.

The governing equations with the corresponding discretization are the main focus of Chapter Three. The use of staggered grids and the hybrid schemes are discussed and the discretization process is described in detail.

Chapter Four describes the solution algorithms implemented in the two transient solvers and in the `D-adaptive solver`. The focus is on the new developments necessary to extend the steady codes `Pamg` and `Pamg-multiphase` to transient capability. The implementation of the `transient Pamg Single-phase` and `transient Pamg Multi-phase` has required a careful consideration of many issues, namely

- The use of Newtons' method in order to accelerate the converging process.
- the use of two automatic differentiation packages[18, 19]and [20].
- the treatment of the non-linearity by line-searching to obtain accurate local Jacobians.
- the implementation of a reinitialization process for the initial guess at each time step.

The `D-adaptive solver` originated some new novel features:

- the definition of interpolation formulas for velocities and volume fractions.
- the establishment of rules for implementing the dimensional interface.

To the author's knowledge, these new features are original in this context.

In Chapter Five the validation process of the `transient Pamg Single-phase` and `transient Pamg Multi-phase` is presented. Some test flow problems of varying degrees of complexity are outlined. The objective is to establish that the solvers give the correct solutions for each of the test cases studied. Comparisons were drawn not with experiments but with the commercial CFD package `CFX 4.3`(see for instance [21]), , and whenever available with the analytic solutions.

Chapter Six is where the `D-adaptive solver` is validated. The interface treatment on the `D-adaptive solver` is outlined with the interpolation formulas presented. Particular emphasis is placed on the asymptotics derived for the one-dimensional Navier-Stokes equations. The methodology for the treatment of the dimensional interfaces on the `D-adaptive solver` is discussed. Particularly difficult domains and constraints are explained. The asymptotic expansion of the

velocities and volume fractions is compared with the computational results for a two-dimensional flow on a channel. Good agreement is observed, demonstrating the correctness of the formulation. Similar asymptotic expansion of the velocities and volume fraction has been formulated in cylindrical coordinates. These expressions are validated with the results for a two-dimensional flow through a pipe. The validation of the `D-adaptive solver` is established in two complex multidomains. The benefits on the CPU-time and amount of work are demonstrated with the simulation by the two-dimensional solver `transient Pamg Multi-phase` and `CFX 4.3` of a fluid flow throw the geometries of the same test cases.

The main conclusions of this thesis are outlined in Chapter Seven. The methodology adopted for treating the coupling between the one-dimensional solver with the two-dimensional one, originating the `D-adaptive solver` has been the principal achievement of this project. However, other important conclusions can not be neglected. The two new transient versions of the originally steady codes `Pamg` and `Pamg-multiphase` have been extremely important since they have been decisive for the steps after. Moreover, the new features that have been developed were implemented in the final solver. Some extra work can be outlined to continue this project. Particularly important can be the design of a new coarse solver for the `transient Pamg Single-phase` and `transient Pamg Multi-phase` which can decrease drastically the computing time, particularly for longer transient simulations. A comparison work on the two automatic differentiation packages, used during this project would be useful for later developments in several fields. Finally the generalisation of the key ideas presented, to other solvers can be extremely important and enlarge the scope of this thesis.

The principal objectives for writing this thesis have been: firstly, to present the new algorithm for solving the time dependent Navier-Stokes equations, secondly to demonstrate that the `D-adaptive solver` simulates correctly the laminar incompressible fluid flows on pipes and thirdly to demonstrate the potential of such approach on the simulation of such flows. The presentation of numerical results takes a relative weight and actually gives a rather distorted image of the work, which has been carried out during the course of this project. The most delicate aspect, has been the development of the software. The necessity of working with several computer codes and packages and the management of several test cases at the same time, has been an extra difficulty.

Chapter 2

Literature Review

2.1 Introduction

In this chapter a literature review of multi-phase flows, numerical schemes, multi-grid methods and adaptivity is presented. The focus will be on the time dependent multi-phase flows, which have been the main concern in this thesis. However, the numerical analysis of several numerical schemes has taken an important part in the design of the transient solvers. Multigrid methods, which represent an important field for Computational Fluid Dynamics, are also outlined. Finally an overview of adaptivity is presented.

Section 2.2 reviews the multi-phase flow models. Particular attention is taken to the derivation of the multi-fluid equations. However, modelling questions, which are of crucial importance to the study of multi-phase flows, are not within the scope of this thesis.

The numerical simulation of multi-phase flows is discussed in Section 2.3 The Navier-Stokes equations are analysed in terms of the challenges that they pose and the constraints that they carry. A brief overview of the Drift-Flux Model is also addressed. Moreover, examples of the many schemes available for the solution of multi-phase flows are described.

Next, the mathematical foundations of the different numerical schemes, which have been proposed for the solution algorithms, are outline in Section 2.4. For a particular case study, implicit, explicit and θ methods are outlined.

Multigrid methods are the subject of Section 2.5. Starting with a historical introduction we go forward to two-level and Full Approximation Storage Schemes. Multigrid cycling strategies are also reviewed.

Section 2.6 presents a review of the known definitions of adaptivity. Some theoretical and experimental work is referenced. The main objective is to launch, in a comparative way, the fundamentals for the dimensional adaptivity, which is the core of the present thesis.

2.2 Basic Flow Models

2.2.1 Multi-Fluid Model

The multi-fluid model is a successful attempt at providing a general framework for the mathematical description of multi-phase equations. Its generalization to the description of multi-phase flows is based on the example of single-phase flow. These are governed by Navier-Stokes equations (see references[2, 22]) which express the conservation of mass, momentum and energy in an Eulerian framework. The multi-fluid equations, like Navier-Stokes equations, express the conservation of mass, momentum and energy using the principles of continuum mechanics. However, the multi-fluid equations do not provide a complete description of a multi-phase flow. They need to be supplied with additional information about the way the phases interact with each other and with themselves. This particular difference, when comparing to the single-flow regime, is crucial for the multi-phase modelling, increasing significantly the complexity of the numerical simulation of multi-phase flows.

The correctness of the multi-fluid model is not absolutely established. The simplifications and approximations that have to be done to get a set of workable equations ignore many details of the flow.

In the basic model, each phase can be described by partial differential equations expressing the conservation of mass, momentum and energy for single-phase flows. The phases are separated by interfaces and at each point which is not on an interface, only the set of governing equations that corresponds to the phase that covers the point, applies.

This is normally known as the microscopic level of description. Normally it is too complex for numerical simulation, and the extra information that it provides is not necessary to model the physics. So multi-fluid equations are derived by averaging the microscopic equations. This process introduces the concept of inter-penetrating continua, which basically postulates that all phases are assumed to be present at each location. Hence conservation laws for mass, momentum and energy are derived for each phase. An extra dependent variable is introduced to characterise each phase in this model: the volume fraction, which is the proportion of volume occupied around each point by a given phase.

Multi-fluid equations, although similar for each phase to the Navier-Stokes equations, are mathematically much more complex because:

- the introduction of the volume fraction as a factor in each term introduces a higher degree of non-linearity of the system of equations.
- the presence of physical constraints on the realisable solutions due to the mandatory variation of the volume fractions (between zero and one).
- the existence of inter-phase transfer terms.

2.2.2 Derivation of the Multi-Fluid Equations

Since this study is only about incompressible fluids, the focus is only on the conservation of mass and momentum. So we do not consider the conservation of energy.

The main stages of the derivation process are presented here in a similar manner as [23], for a two-phase flow. However, it can be easily extended to multi-phase flows. At a microscopic level of description, a two-phase flow is a set of two mutually exclusive regions in which a single-phase flow occurs. So, single-phase governing equations are assumed to apply at each point. However, at the phase interfaces jump conditions are applied.

The equations of motion for each phase are:

- Conservation of mass:

$$\frac{\partial \rho}{\partial t} + \nabla.\rho\mathbf{v} = 0 \tag{2.1}$$

- Conservation of linear momentum:

$$\frac{\partial \rho\mathbf{v}}{\partial t} + \nabla.\rho(\mathbf{v}\otimes\mathbf{v}) = \nabla.\mathbf{T} + \rho f \tag{2.2}$$

- Conservation of angular momentum:

$$\mathbf{T} = \mathbf{T}^T \tag{2.3}$$

Here ρ denotes the density, $\mathbf{v}$ the velocity, $\mathbf{T}$ the stress tensor, and f the body force density.

At the interface, the jump conditions are

jump condition for mass

$$\|\rho(\mathbf{v} - \mathbf{v}_{int}).\mathbf{n}\| = 0 \tag{2.4}$$

jump condition for momentum

$$\|\rho\mathbf{v}(\mathbf{v} - \mathbf{v}_{int}).\mathbf{n} - \mathbf{T}.\mathbf{n}\| = \sigma\kappa\mathbf{n} \tag{2.5}$$

Where $\mathbf{v}_{int}$ is the velocity of the interface, σ is the surface tension, κ the average curvature of the interface, and $\mathbf{n}$ is the unit normal, oriented in such a way that if $\|f\| = f^\beta - f^\alpha$ then $\mathbf{n}$ points from phase α to phase β.

Let $<>$ denote an averaging process, so that if $f(x,t)$ is an exact microscopic field the $< f > (x,y)$ is the corresponding averaged field. Some common averaging operator can be found in [23, see pages 268-269]. The averaging operators are assumed to satisfy Reynolds' rules, Leibnitz's rule and Gauss' rule[23, see the definitions in page 269].

Introducing the phase function $X_\alpha(\mathbf{x},t)$ defined as

$$X_\alpha(\mathbf{x},t) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is in phase } \alpha \text{ at time } t \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

Noting that

$$X_\alpha \frac{\partial \rho}{\partial t} = \frac{\partial}{\partial t} X_\alpha \rho - \rho \frac{\partial X_\alpha}{\partial t} = \frac{\partial}{\partial t} X_\alpha \rho + \rho \mathbf{v}_{int} . \nabla X_\alpha$$

and

$$X_\alpha \nabla . \rho \mathbf{v} = \nabla . (X_\alpha \rho \mathbf{v}) - \rho \mathbf{v} . \nabla X_\alpha$$

we have

$$\frac{\partial}{\partial t} < X_\alpha \rho > + \nabla . < X_\alpha \rho \mathbf{v} > = < [\rho(\mathbf{v} - \mathbf{v}_{int})]_\alpha . \nabla X_\alpha > \qquad (2.7)$$

Similar treatment for the momentum equations yields:

$$\frac{\partial}{\partial t} < X_\alpha \rho \mathbf{v} > + \nabla . < X_\alpha \rho \mathbf{v} \otimes \mathbf{v} > = \nabla . < X_\alpha T > + < X_\alpha \rho f > + < [\rho(\mathbf{v} - \mathbf{v}_{int}) - T]_\alpha . \nabla X_\alpha > \qquad (2.8)$$

The terms

$$< [\rho(\mathbf{v} - \mathbf{v}_{int})]_\alpha . \nabla X_\alpha > = \Gamma_\alpha$$

and

$$< [\rho(\mathbf{v} - \mathbf{v}_{int}) - T]_\alpha . \nabla X_\alpha > = \mathbf{M}_\alpha$$

are the interfacial source terms.

It can be shown, using the jump conditions (2.4) and (2.5), that:

$$\sum_{\alpha=1}^{2} \Gamma_\alpha = 0$$

and

$$\sum_{\alpha=1}^{2} \mathbf{M}_\alpha = \mathbf{M}_m$$

The first relationship expresses the fact that the total mass in the system is conserved, while the second means that momentum balances must take into account inter-facial forces and particularly, the surface tension.

The volume fraction (also called volumetric concentration or relative resistance time) of phase α is defined by:

$$r_\alpha = < X_\alpha >$$

The phasic average of the variable ϕ is defined by

$$\tilde{\phi} = \frac{<X_\alpha \phi>}{r_\alpha}$$

and the mass-weighted average of the variable ψ is defined by

$$\hat{\psi} = \frac{<X_\alpha \rho \psi>}{r_\alpha \tilde{\rho_\alpha}}$$

It is convenient to write the stresses $\tilde{T}_\alpha$ in terms of the pressures plus extra stresses getting:

$$\tilde{\mathbf{T}}_\alpha = -\tilde{p_\alpha}\mathbf{I} + \tilde{\tau_\alpha}$$

where $\tilde{\tau_\alpha}$ is the stress tensor

Similarly, the inter-facial momentum transfer term M_α is usually separated into various parts:

$$\mathbf{M}_\alpha = \Gamma_\alpha \mathbf{v}_{\alpha,int} - p_{\alpha,int} \nabla r_\alpha + \mathbf{M}_\alpha^d$$

where $\mathbf{M}_\alpha^d$ is the inter-facial force density,$p_{\alpha,int}$ is the inter-facial pressure for phase α and $\mathbf{v}_{\alpha,int}$ is the inter-facial velocity for phase α.

So the averaged equations (2.7) and (2.8) become:

$$\frac{\partial r_\alpha \tilde{\rho_\alpha}}{\partial t} + \nabla.(r_\alpha \tilde{\rho_\alpha} \hat{\mathbf{v}_\alpha}) = \Gamma_\alpha \tag{2.9}$$

$$\frac{\partial r_\alpha \tilde{\rho_\alpha} \hat{\mathbf{v}_\alpha}}{\partial t} + \nabla.(r_\alpha \tilde{\rho_\alpha} \hat{\mathbf{v}_\alpha} \otimes \hat{\mathbf{v}_\alpha}) = -r_\alpha \tilde{\mathbf{T}} + \mathbf{M}_\alpha \tag{2.10}$$

Noting that different averaging have been applied to different quantities and the body force has been neglected.

A simplified version of the multi-fluid equations can be found in [24]. For each phase, and for incompressible flows, it takes the following form:

- Continuity Equation :

$$\frac{\partial r_\alpha \rho_\alpha}{\partial t} + \nabla . r_\alpha \rho_\alpha \mathbf{v}_\alpha = \Gamma_\alpha \tag{2.11}$$

- Momentum Equation:

$$\frac{\partial r_\alpha \rho_\alpha \mathbf{v}_\alpha}{\partial t} + \nabla.(r_\alpha \rho_\alpha \mathbf{v}_\alpha \otimes \mathbf{v}_\alpha) = -r_\alpha \nabla p_\alpha + \nabla . r_\alpha (\mathbf{T}_\alpha + \mathbf{T}_\alpha^t) + r_\alpha \rho_\alpha g + \mathbf{v}_{\alpha,int} \Gamma_\alpha + \mathbf{M}_{int,\alpha} - \nabla r_\alpha . \mathbf{T}_{int}. \tag{2.12}$$

Where Γ_α is the mass transfer to phase α, $\mathbf{M}_{int,\alpha}$ is the interfacial drag, $\mathbf{T}_\alpha$ and $\mathbf{T}_\alpha^t$ are the laminar and turbulent shear stress tensors.

The Homogeneous Model

In this model, the momentum equations are summed over all phases to give a single momentum transport equation of the form:

$$\frac{\partial}{\partial t}(\rho v) + \nabla.(\rho(\mathbf{v} \otimes \mathbf{v}) - \mu\mathbf{T}) = -\nabla p + \rho f \tag{2.13}$$

Where ρ and μ are the density and viscosity of the mixture:

$$\rho = \sum_{\alpha=1}^{M} r_\alpha \rho_\alpha$$

$$\mu = \sum_{\alpha=1}^{M} r_\alpha \mu_\alpha$$

The continuity equations remain distinct, allowing the volume fractions to be determined. The energy equations if present can also be added together as the momentum equations. See [25] for a more complete cover of this homogeneous model.

Normally the homogeneous model is applicable if the flow is strongly stratified under the influence of the gravity or if the velocities and energy for each phase equalise very quickly, due to large momentum and energy transfer terms.

The Drift Flux Model

In this model a total momentum equation is considered, together with a mass conservation equation for each phase.

The principal advantages of this model [26, page 744] are:

- the equations are in conservative form, which makes it easier to discretize by finite volume methods;
- the interfacial friction term is cancelled out in the momentum equation, although it appears in an additional algebraic relation called the slip law;
- one does not have to work a third boundary condition at the inlet;
- it is generally hyperbolic, depending on the form of the slip law.

Following [27] we have the following governing equations for a two-phase flow:

$$\frac{\partial}{\partial t}(\rho_\alpha r_\alpha x_j + \rho_\beta r_\beta y_j) + \frac{\partial}{\partial x}(\rho_\alpha r_\alpha \mathbf{v}_\alpha x_j + \rho_\beta r_\beta \mathbf{v}_\beta y_j) = 0 \qquad j = 1...N \tag{2.14}$$

$$\frac{\partial}{\partial t}(\rho_\alpha r_\alpha \mathbf{v}_\alpha + \rho_\beta r_\beta \mathbf{v}_\beta) + \frac{\partial}{\partial x}(\rho_\alpha r_\alpha \mathbf{v}_\alpha^2 + \rho_\beta r_\beta \mathbf{v}_\beta^2 + P) = T^w - \rho g \sin\theta \tag{2.15}$$

Where r_α and r_β are the volume fractions, $\mathbf{v}_\alpha$ and $\mathbf{v}_\beta$ are the velocities, x_j (resp.y_j) are the mass fraction of component j in phase α (resp. β), p is the pressure, $\rho = \rho_\alpha r_\alpha + \rho_\beta r_\beta$ is the mixture density, g is the gravity acceleration and T^w is the wall friction.

To close the system, normally are used the following equations:

- $$r_\alpha + r_\beta = 1, \sum_{j=1,2} x_j = 1, \sum_{j=1,2} y_j = 1. \tag{2.16}$$

- Thermodynamic closure laws: they give the fluid and component properties $\Gamma(P, T, c_{j(j=1...2)})$, where c_j is the jth component mass fraction.
- Hydrodynamic closure law: it can be written formally as:

$$\phi(\mathbf{V}_M, x_\beta, \Gamma(P, T, c_j, (j = 1...2), \delta\mathbf{V}, x) = 0 \tag{2.17}$$

where $\mathbf{V}_M = (\rho_\alpha r_\alpha \mathbf{v}_\alpha + \rho_\beta r_\beta \mathbf{v}_\beta)/\rho$ is the center of mass velocity, $x_\beta = \rho_\beta r_\beta / \rho$ is the beta mass fraction, and $\delta\mathbf{V} = \mathbf{v}_\beta - \mathbf{v}_\alpha$ is the slip velocity between the two phases.

The equations (2.14) and (2.15) can be written in the following conservative form:

$$\frac{\partial}{\partial t}\mathbf{W}(x,t) + \frac{\partial}{\partial x}F(x, \mathbf{W}(x,t)) = Q(x, \mathbf{W}(x,t)) \tag{2.18}$$

Where $\mathbf{W}^t = (\rho_\alpha r_\alpha x_j + \rho_\beta r_\beta y_j, j = 1...2, \rho_\alpha r_\alpha \mathbf{v}_\alpha + \rho_\beta r_\beta \mathbf{v}_\beta)$. Given $\mathbf{W}$ and x, F and Q can be computed solving the closure laws but there is no analytical expression of $F(x, \mathbf{W})$ and $Q(x, \mathbf{W})$. The dependence of F with respect to x comes from the fact that the hydrodynamic closure law (2.17) depends on x.

Since there is no analytical expression for $F(x, \mathbf{W})$, it is difficult to study the properties of the System (2.18). Numerical tests[27] show that, for usual values of $\mathbf{W}$, the eigenvalues are real: $\lambda_1 < \lambda_2 \leq ... \leq \lambda_N < \lambda_{N+1}$ with $\lambda_1 < 0, \lambda_{N+1} > 0$. The system is then hyperbolic and non-linear.

Another important characteristic of the model from the numerical point of view is that the first and last eigenvalues are 10-100 time bigger than the other ones. From a physical point of view, this large eigenvalues are associated with "pressure waves" and can therefore be very large, whereas the others are associated to "composition fraction waves" and are of the order of the fluid velocities.

2.3 Numerical Simulation of Multi-Phase Flows

The numerical simulation of multi-phase flows has been object of study in many books and papers. See for instance reference [6] for a detailed description. In the case of multiphase flows, three fundamental issues need to be addressed:

1. The governing equations can be ill-posed (see for instance[23], complemented with[3]). This means that the Jacobian of the flux function, with respect to the conservative variables, can have complex eigenvalues for certain regions of phase space. In this case their numerical solutions will therefore tend to exhibit non-physical oscillations, since the amplitude of all error modes increases with time over the entire range of frequencies. To handle this problem, usually a sufficient amount of numerical diffusion is added by making use of upwind diferencing. However, this can blur the solution features since the schemes are then first order accurate in space. So, the reformulation of the multi-fluid models to eliminate the ill-posedness of the equations is an important topic of research.

2. Normally, the flux function is not available and so the conservative form of the governing equations can not be rewritten. This difficult the implementation of numerical schemes, since most of them are only available for the conservative form of equations.

3. In many multi-phase flows, phenomena of interest propagate at a very small velocity, when compared to the speed of sound. Explicit methods (see Section 2.4), which are the simplest way of providing accurate numerical solutions, are therefore penalised due to their stability limit. So, a large number of very small explicit steps are necessary to model such a slow phenomenon. In this case, implicit methods are an important alternative. The adequate implicit/explicit methods choice is also an important area of research. Semi-implicit methods [28] have been a possible approach to remove the stability constraints. The convective terms are treated explicitly, while the inter-phase coupling and the pressure terms, which are responsible for the CFL (Courant, Friedrichs and Lewy) limitation (see for instance[4]) are treated implicitly.

Having discussed some of the issues in the simulation of multi-phase flows, we now present some representative algorithms, developed with different techniques.

The Simple Algorithm

The acronym SIMPLE stems from semi-implicit method for pressure-linked equations. A SIMPLE algorithm, in its original step-by-step procedure, can be described as follow[29]:

1. Given approximations u^n, v^n and p^n of the velocity and pressure field, solve the momentum equations implicitly to obtain a new approximate velocity solution u^* and v^*.
2. Compute a pressure correction Δp so that the velocity fields $u^{n+1} \equiv u^* + u^{(cor)}$ and $v^{n+1} \equiv v^* + v^{(cor)}$ satisfy the continuity equation. $u^{(cor)}$ and $v^{(cor)}$ are the velocity corrections associated with the pressure correction. Once Δp is know, $u^{(cor)}$ and $v^{(cor)}$ can be computed.
3. Update the velocities: $u^{n+1} \equiv u^* + u^{(cor)}$ and $v^{n+1} \equiv v^* + v^{(cor)}$.
4. Update the pressures: $p^{n+1} \equiv p^n + \Delta p$.
5. With this new pressure values, repeat steps 1 to 5 until convergence is achieved.

So SIMPLE is an example of a segregated solver. It computes the velocities and the pressures separately relying on global linearisations.

IPSA - Inter Phase Slip Algorithm

The IPSA method is a well-known implicit method for the solution of transient multi-fluid equations. Like SIMPLE it is a segregated solver. Basically is an extension of the SIMPLE algorithm to multi-phase flows. It is used in the commercial codes `CFX 4.3`[21] and `fluent`[30]. The original IPSA-PEA method as used in `fluent`[30] is described as:

- The equations are solved on a staggered grid and discretised using a finite volume approach.
- Given an approximate solution on each cell, each equation is solved in turn to give a correction for a particular value. Volume fractions and velocity corrections are obtained from the continuity and momentum equations. Finally the pressure correction is computed from the overall continuity equation, which is obtained by adding all the continuity equations together. This overall equation is manipulated to obtain a linear relationship between the pressure correction at one point and its nearest neighbours. This relationship uses the differential coefficients of velocity variations with pressure obtained from the momentum equations. If the energy is included, then the corresponding equations lead to enthalpy corrections.
- Partial Elimination Algorithm (PEA) is introduced to take into account the strong local coupling between the phases and so accelerating convergence. It consists of algebraic manipulations of the momentum equations so that the velocities for each phase are eliminated from the momentum equations for each phase.
- In a single IPSA cycle, values are corrected following the order:

1. enthalpy;
2. volume fractions;
3. velocities;
4. pressure.

Coupled Solvers for the Navier-Stokes Equations

One fundamental difficulty, which one meets when solving the transient Navier-Stokes equations for fluid flows, is that there is a strong coupling between the pressure field and the velocity fields via the continuity equation. The SIMPLE family solvers linearize the equations over the domain to obtain a relationship between pressure corrections and velocity corrections. However, this linearisation is often of poor quality and consequently the convergence rate of the method is slow.

Another important issue is that some of the derivatives, which form the entries of the Jacobian, can take large values. In such situations, segregated solvers tend to over-correct some values, since the flow variables are closely coupled together and information about the Jacobians is not incorporated. For instance a correction from one equation and its effect on the other equations is ignored, if the derivative take a large value. We can resume this behaviour saying that the correction to one equation throws the others off-balance. To minimise this problem the computations are under-relaxed so that the correction steps are reduced. However the cost associated to this procedure is that the convergence rates often degrade.

To overcome these difficulties the coupled solvers methodology open a new approach[31, 32, 33]. Usually the equations are discretised on a staggered grid using a finite volume approach. The momentum and continuity equations, written on a group of cells, are solved simultaneously by Newton's method. These equations form a non-linear subsystem for the vector of unknowns written on the group of cells. However it is necessary to have an approximation of the Jacobian in order to apply Newton's method.

If the equations for the entire domain are solved simultaneously, the Jacobian matrices are large and their degree of sparseness is important for performance.

However, another approach can be though of. The idea is to couple the system written on a single cell of the computational domain[31, 15]. In this case, only local coupling is used and the subsystems are much smaller, 5x5 for a single-phase two-dimensional Navier-Stokes solver. So, an iterative process based on that block of equations of each subsystem is used to solve the complete set of equations of the entire system. These are normally solved in a Gauss-Seidel manner. This smoothers usually known as Vanka-type smoothers[34], have been object of study in several papers (see for instance [35, 36]). The performance of `Pamg` suggests that it is the local coupling, which is crucial for convergence rates and robustness (see[15]).

More recent work has been done on solving the coupled systems in a line-wise fashion[37]. The coupled method is implemented in a way that the variables

belonging to entire lines of cells are updated simultaneously. Some variations on the order of updating the variables originate differently implemented algorithms. The SCGS/LS[38] solves only for the pressure along gridlines and this appeared to be more efficient than the cell-wise approach for the driven cavity problem. The SCAL(Symmetrically Coupled Alternating Line)[39] and the CLGS(Collective line Gauss Seidel)[40] algorithms are other possible implementations. SCAL differs from CLGS in the sweeping pattern used to produce updates. Alternating zebra in the former and line-Gauss-Seidel in the latter.

In several implementations, the basic coupled solver is usually applied with a multigrid method. This combination forms an efficient iterative solver. However, two important issues must be solved:

- The linearisation must be of sufficiently good quality to rapidly eliminate the high frequencies of the error. Otherwise it will not perform well as a multigrid smoother.
- The coarse grid correction has to be efficiently computed. In fact, the multigrid method accelerates the convergence of the single grid solver since it provides a framework in which the global coupling can be quickly taken into account, using the coarse grid computations.

2.4 Numerical Schemes

2.4.1 Introduction

In this section we shall be concerned with the numerical solution of partial differential equations. The focus will be on some algorithms that can be used for solving a model problem. Choosing a very simple model we start with an explicit difference method, which is very straightforward in use, and establish the main issues that it rises. As we shall show, however, the numerical solution becomes unstable unless the time step is severely restricted. So we shall go on to consider other, more elaborate numerical methods which can avoid the time step restriction. This subject is covered in detail in several books and articles. For a complete overview of each method see references[4, 5, 41]. Other literature covers particular applications. For a study of implicit, explicit operators for the upwind differencing method on the Euler and Navier-Stokes equations (see [42]). More recent work is for instance reference[35].

2.4.2 Explicit Scheme. Truncation Error

Many problems in science and engineering are modelled by special cases of the linear parabolic equation[4], in the unknown $u(x,t)$:

$$e(x,t)\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(a(x,t)\frac{\partial u}{\partial x}) + b(x,t)\frac{\partial u}{\partial x} + c(x,t)u + d(x,t) \qquad (2.19)$$

where e and a are strictly positive. Let $u(x,0) = u^0(x)$, where $u^0(x)$ is a given function, be the initial condition. The solution of the problem will be required to satisfy Equation (2.19) for $t > 0$ and x in an open region $\mathcal{R}$ which can be the whole real line, the half-line $x > 0$, or an interval such as $(0,1)$. Supposing that we have Dirichlet boundary conditions (i.e. the solution is given to be zero at each end of the range, for all values of t), and after changing to dimensionless variables this problem becomes: find $u(x,t)$ defined for $x\epsilon[0,1]$ and $t \geq 0$ such that

$$u_t = u_{xx} \qquad for\ t > 0,\ 0 < x < 1, \tag{2.20}$$

$$u(0,t) = u(1,t) = 0 \quad for\ t > 0, \tag{2.21}$$

$$u(x,0) = u^0(x) \qquad for\ 0 \leq x \leq 1. \tag{2.22}$$

where the common subscript notation denote partial derivatives.

To approximate Equation (2.20) by finite differences we divide the domain $\mathcal{R} \times [0, t_F]$ by a set of lines parallel to the $x-$ axis and $t-$ axis to form a grid mesh. For simplicity only, we assume that the grid mesh is uniform, meaning that the set of lines is equally spaced. Writing Δx and Δt for the line spacings, the crossing points

$$(x_j = j\Delta x,\ t_n = n\Delta t),\ j = 0, 1, ..., J, n = 0, 1, ..., \tag{2.23}$$

where

$$\Delta x = 1/J \tag{2.24}$$

are called the grid points or mesh points. We are looking for approximations

$$U_j^n \approx\ u(x_j, t_n) \tag{2.25}$$

of the solution at these grid points. For Equation (2.20) the simplest difference scheme based at the mesh point (x_j, t_n) uses a forward difference for the time derivative; this gives

$$\frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{\Delta t} \approx \frac{\partial u}{\partial t}(x_j, t_n) \tag{2.26}$$

for any function u with a continuous $t-$ derivative. The scheme uses a centred second difference for the second order space derivative:

$$\frac{u(x_{j+1}, t_n) - 2u(x_j, t_n) + u(x_{j-1}, t_n)}{(\Delta x)^2} \approx \frac{\partial^2 u}{\partial x^2}(x_j, t_n). \tag{2.27}$$

Equating the left-hand sides of these two equations we get

$$U_j^{n+1} = U_j^n + \nu(U_{j+1}^n - 2U_j^n + U_{j-1}^n) \tag{2.28}$$

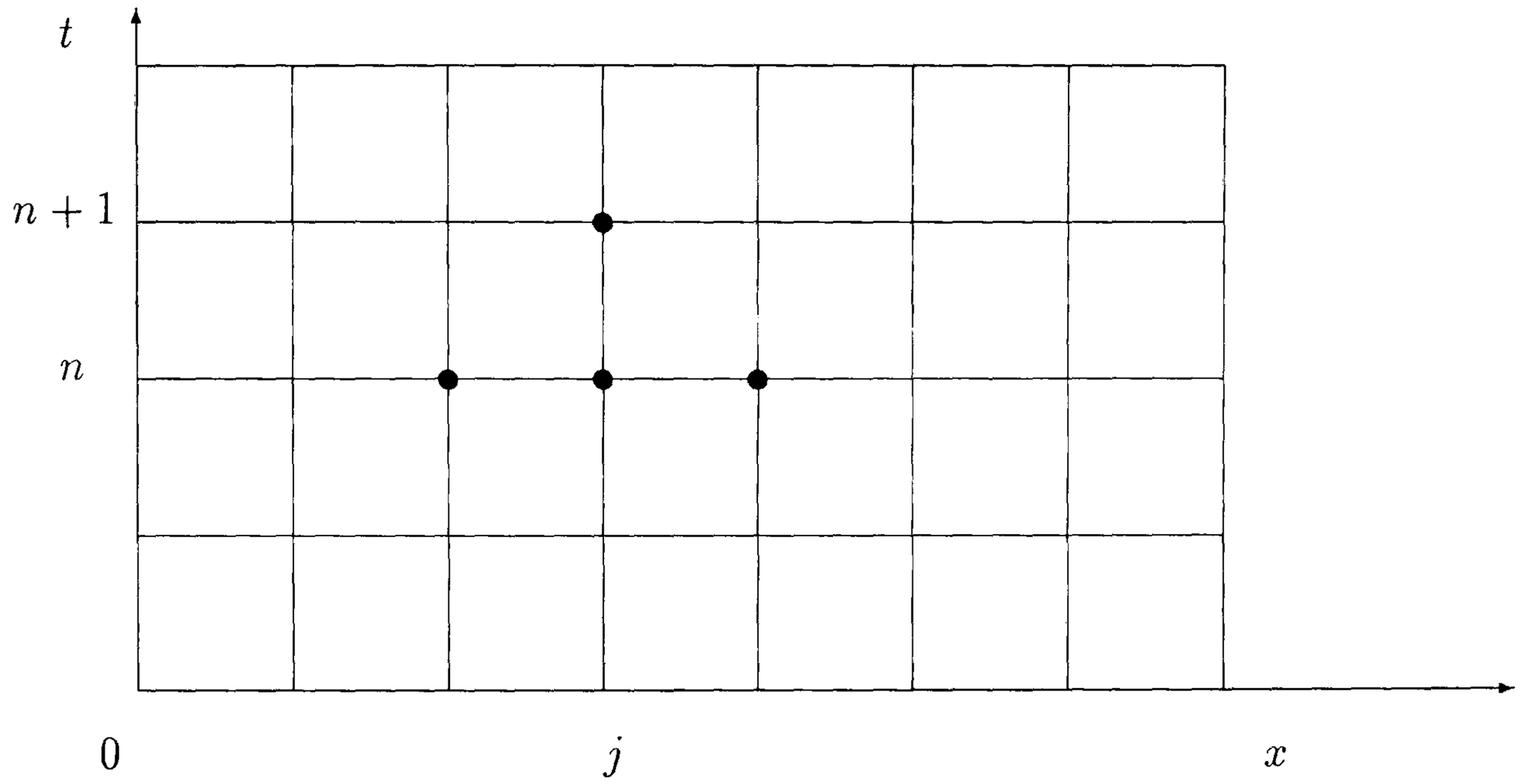


Figure 2.1: An explicit scheme. t_{n+1} is calculated from t_n

where

$$\nu = \frac{\Delta t}{(\Delta x)^2} \tag{2.29}$$

The pattern of grid points involved in Equation (2.28) is shown in Figure 2.1. At each time level t_{n+1} can be independently calculated from values at time level t_n. For this reason this is called an explicit difference scheme.

Before proceeding let us remember some definitions and notation. Defining the finite differences in the same way in the two variables t and x, there are three kinds of finite differences:

- Forward differences

$$\Delta_{+t}u(x,t) = u(x,t+\Delta t) - u(x,t), \tag{2.30}$$

$$\Delta_{+x}u(x,t) = u(x+\Delta x,t) - u(x,t). \tag{2.31}$$

- Backward differences

$$\Delta_{-t}u(x,t) = u(x,t) - u(x,t-\Delta t), \tag{2.32}$$

$$\Delta_{-x}u(x,t) = u(x,t) - u(x-\Delta x,t). \tag{2.33}$$

- Central differences

$$\delta_t u(x,t) = u(x,t+\frac{1}{2}\Delta t) - u(x,t-\frac{1}{2}\Delta t), \tag{2.34}$$

$$\delta_x u(x,t) = u(x+\frac{1}{2}\Delta x,t) - u(x-\frac{1}{2}\Delta x,t). \tag{2.35}$$

We can now define the truncation error of Equation (2.28). Using the equivalent formula:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta x)^2} \tag{2.36}$$

The truncation error is then the difference between the two sides of the equation, when the approximation U_j^n is replaced throughout by the exact solution $u(x_j, t_n)$ of the differential equation. We can define the truncation error $\tau(x,t)$ by:

$$\tau(x,t) = \frac{\Delta_{+t} u(x,t)}{\Delta t} - \frac{\delta_x^2 u(x,t)}{(\Delta x)^2} \tag{2.37}$$

so that

$$\tau(x,t) = (u_t - u_{xx}) + (\frac{1}{2}u_{tt}\Delta t - \frac{1}{12}u_{xxxx}(\Delta x^2) + \ldots = \frac{1}{2}u_{tt}\Delta t - \frac{1}{12}u_{xxxx}(\Delta x)^2 + \ldots \tag{2.38}$$

where the leading terms are called the principal part of the truncation error, and u satisfies the differential equation. It is often convenient to truncate the infinite Taylor series, introducing a remainder term, for example

$$u(x, t+\Delta t) = u(x,t) + u_t\Delta t + \frac{1}{2}u_{tt}(\Delta t)^2 + \frac{1}{6}u_{ttt}(\Delta t)^3 + \ldots = u(x,t) + u_t\Delta t + \frac{1}{2}u_{tt}(x,\eta)(\Delta t)^2, \tag{2.39}$$

where $\eta \, \epsilon \, (t, t+\Delta t)$.

Doing the same thing for the x expansion the truncation error becomes

$$\tau(x,t) = \frac{1}{2}u_{tt}(x,\eta)\Delta t - \frac{1}{12}u_{xxxx}(\xi,t)(\Delta x)^2 \tag{2.40}$$

where $\xi \, \epsilon \, (x - \Delta x, x + \Delta x)$, from which it follows that

$$\mid \tau(x,t) \mid \leq \frac{1}{2}M_{tt}\Delta t + \frac{1}{12}M_{xxxx}(\Delta x)^2 \tag{2.41}$$

or

$$\mid \tau(x,t) \mid \leq \frac{1}{2}\Delta t[M_{tt} + \frac{1}{6\nu}M_{xxxx}] \tag{2.42}$$

where M_{tt} is a bound for $\mid u_{tt} \mid$ and M_{xxxx} is a bound for $\mid u_{xxxx} \mid$.

This equation tells us that for a fixed ratio ν, $\mid \tau \mid$ will behave asymptotically like $\mathcal{O}(\Delta t)$ as $\Delta t \to 0$. Except for special values of ν this will be the highest power of Δt for which such a statement could be made, so that the scheme is first order

accurate. However, since u satisfies $u_t = u_{xx}$ everywhere, we also have $u_{tt} = u_{xxxx}$ and hence

$$\tau(x,t) = \frac{1}{2}(1 - \frac{1}{6\nu})u_{xxxx}\Delta t + \mathcal{O}((\Delta t)^2). \tag{2.43}$$

So for $\nu = \frac{1}{6}$ the scheme is second order accurate. However this is a rather special case. The explicit scheme is convergent if

$$\nu \leq \frac{1}{2}. \tag{2.44}$$

(proof of this can be found in [4, pages 15-17]).

2.4.3 Implicit Scheme

The stability limit Equation (2.44) which is equivalent to

$$\Delta t \leq \frac{1}{2}(\Delta x)^2 \tag{2.45}$$

can be very restrictive. In fact it implies that many time steps will be necessary to reach a solution over a reasonably large time interval. Moreover, if we have to reduce Δx to improve the accuracy of the solution, the amount of work involved will increase very quickly, since we shall have to reduce Δt as well.

For the reasons outlined above, we can in Equation (2.36) replace the forward time difference by the backward time difference(the space difference remaining the same), and get the implicit scheme:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{(\Delta x)^2} \tag{2.46}$$

This can be written using the difference notation given in Section 2.4.2 as

$$\Delta_{-t} U_j^{n+1} = \nu \delta_x^2 U_j^{n+1} \tag{2.47}$$

where ν is given by Equation (2.29). As we can see on Figure 2.2 the scheme involves three unknown values of U on the new time level $n+1$.

We must write Equation (2.47) in the form

$$-\nu U_{j-1}^{n+1} + (1+2\nu)U_{j-1}^{n+1} - \nu U_{j+1}^{n+1} = U_j^n. \tag{2.48}$$

and solve this system of equations simultaneously.

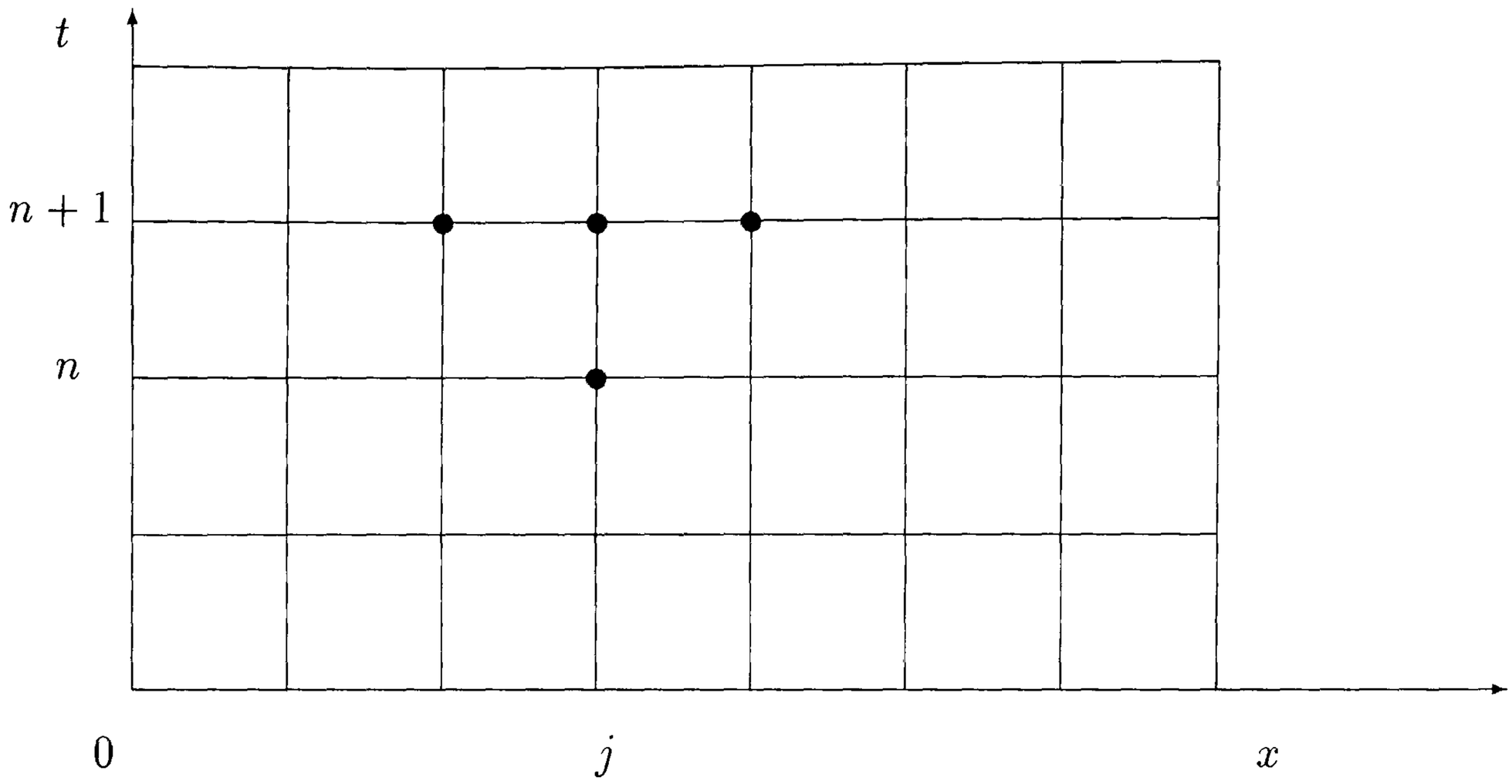


Figure 2.2: An implicit scheme. Three unknown values on t_{n+1} are involved

2.4.4 Theta Scheme

So far we have considered two finite difference methods, which differ only in that one approximates the second derivative by three points on the old time level,t_n, and the other uses the three points on the new time level, t_{n+1}. A natural generalisation is to an approximation which uses all six points, which can be regarded as taking a weighted average of the two formulae. We obtain the six-point scheme (see Figure 2.3)

$$U_j^{n+1} - U_j^n = \nu[\theta\delta_x^2 U_j^{n+1} + (1-\theta)\delta_x^2 U_j^n], \qquad j = 1, 2, ..., J-1. \tag{2.49}$$

Assuming that we are using an average with non-negative weights, so that

$$0 \leq \theta \leq 1,$$

the particular cases of $\theta = 0$ and $\theta = 1$ give the explicit and implicit schemes respectively. For $\theta = 1/2$ we have the Crank-Nicolson scheme.

For any $\theta \neq 0$ we have the tridiagonal system to solve for U_j^{n+1}:

$$-\theta\nu U_{j-1}^{n+1} + (1+2\theta\nu)U_j^{n+1} - \theta\nu U_{j+1}^{n+1} = [1 + (1-\theta)\nu\delta_x^2]U_j^n \tag{2.50}$$

This system can be solved for instance by Thomas algorithm, since the matrix of the coefficients is diagonally dominant (see [4, pages 23-25]).

We can study the stability of this scheme using Fourier analysis. Substituting the form $U_j^n = (\lambda)^n \epsilon^{ik(j\Delta x)}$ into Equation (2.50) and after some algebraic manipulation we get

$$\lambda = \frac{1 - 4(1-\theta)\nu \sin^2 \frac{1}{2} k\Delta x}{1 + 4\theta\nu \sin^2 \frac{1}{2} k\Delta x}. \tag{2.51}$$

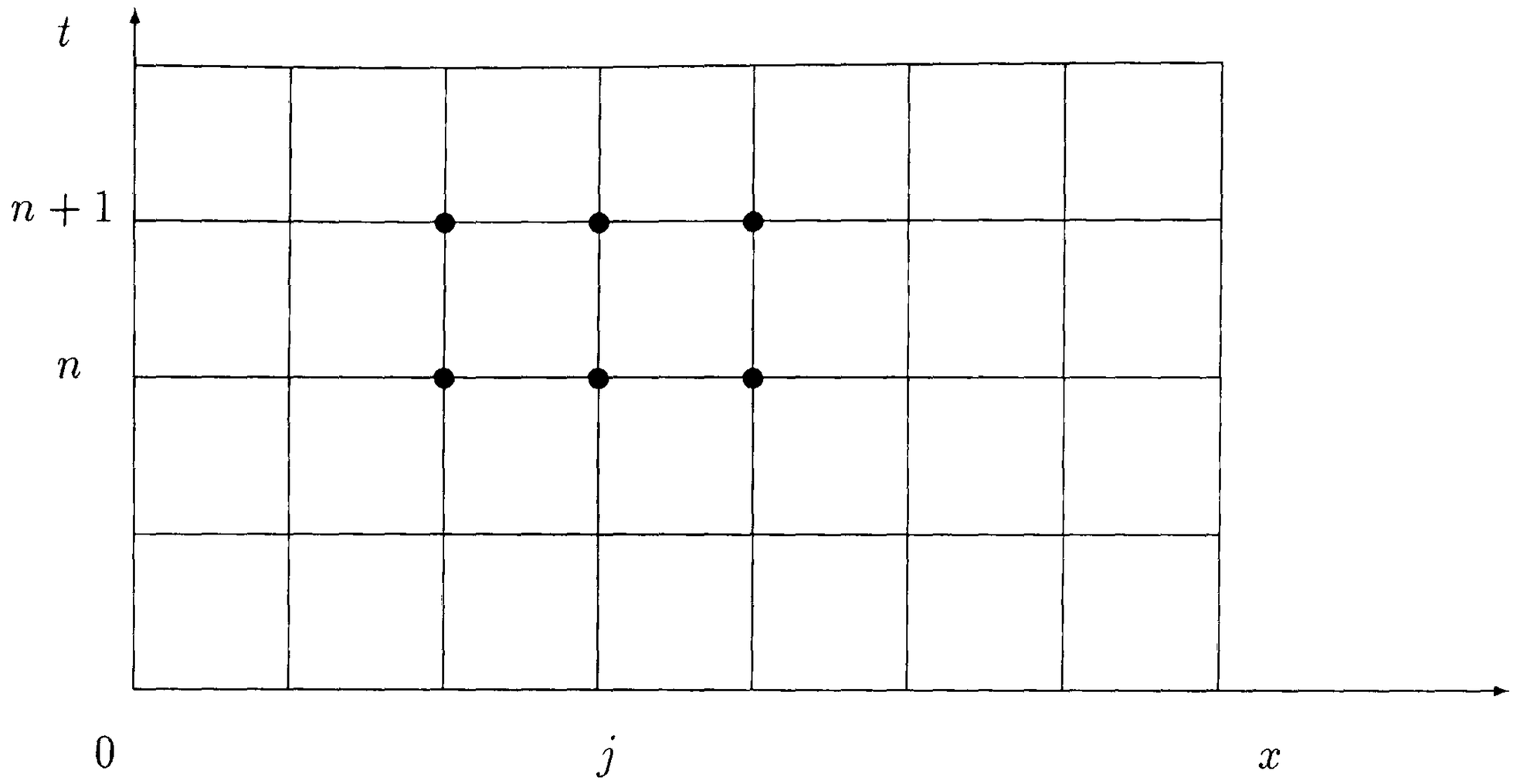


Figure 2.3: The θ scheme. Three points on t_{n+1} and three points on t_n are used for the approximation

Since $\nu > 0$, and we are assuming that $0 \leq \theta \leq 1$, we can never have $\lambda > 1$. So instability arises when $\lambda < -1$, that is when

$$4\nu(1-2\theta)\sin^2\frac{1}{2}k\Delta x > 2. \tag{2.52}$$

The mode most liable for instability is the one for which the left side is largest, which is when $k\Delta x = \pi$. So, this is an unstable mode when

$$\nu(1-2\theta) > \frac{1}{2}. \tag{2.53}$$

This inequality shows the stability condition of the earlier explicit scheme ($\theta = 0$), and that for the implicit scheme ($\theta = 1$) it is not unstable for any value of ν.

Hence, we can summarise the necessary and sufficient conditions for the stability of the θ scheme Equation (2.49) as

$$\left|\begin{array}{l} 0 \leq \theta < 1/2, \quad stable\ if\ and\ only\ if\ \nu \leq 1/2(1-2\theta)^{-1} \\ 1/2 \leq \theta \leq 1, \quad stable\ for\ all\ \nu \end{array}\right. \tag{2.54}$$

The two cases are often referred to as conditional and unconditional stability respectively. As soon as θ is non-zero a tridiagonal system has to be solved, so there would seem to be no advantage in using schemes with $0 < \theta < 1/2$ which are only conditionally stable, unless they are more accurate. It can be demonstrated (see[4, pages 27-28]) that the truncation error of the θ scheme is given by:

$$\begin{aligned} \tau_j^{n+1/2} = {} & [u_t - u_{xx}] + [(1/2-\theta)\Delta t u_{xxt} - 1/12(\Delta x)^2 u_{xxxx}] \\ & + [1/24(\Delta t)^2 u_{ttt} - 1/8(\Delta t)^2 u_{xxtt}] \\ & + [1/12(1/2-\theta)\Delta t(\Delta x)^2 u_{xxxxt} - 2/6!(\Delta x)^4 u_{xxxxxx}] \end{aligned} \tag{2.55}$$

The second term in Equation (2.55) shows that we shall normally have first order in Δt. However, if we have $\theta = 1/2$, the Crank-Nicolson scheme, then is second order accurate in both Δx and Δt. This important property, let us take larger time steps, for example $\Delta x = \mathcal{O}(\Delta t)$. Then the truncation error is $\mathcal{O}(\Delta t^2)$ and we can achieve good accuracy in a more economic way.

Another possible choice is to choose θ, Δt and Δx in order to eliminate the second term of Equation (2.55). This takes us to the relation:

$$\nu = \frac{1}{6(1-2\theta)} \tag{2.56}$$

However this relationship requires $(\Delta x)^2 \leq 6\Delta t$ to ensure that $\theta \geq 0$. In the case of $\theta = 0$, explicit scheme, $\nu = 1/6$, as we have show in Section 2.4.2.

2.5 Multigrid Methods

2.5.1 Introduction

Multigrid methods have been developing rapidly over the last years and are now a powerful tool for the efficient solution of elliptic and hyperbolic equations. Needless to say, elliptic and hyperbolic partial differential equations are, by and large, at the heart of most mathematical models used in engineering and physics, giving rise to extensive computations. Often the problems that one would like to solve exceed the capacity of even the most powerful computers, or the time required is too great to allow inclusion of advanced mathematical models in the design process of technical apparatus, from microchips to aircraft, making design optimisations more difficult. Multigrid methods are a prime source of important advances in algorithmic efficiency, finding a rapidly increasing number of users. Table 2.1, based on the multigrid bibliography in[43], illustrates the rapid growth of the multigrid literature, a growth which has continued unabated since 1985.

Year	64	66	71	72	73	75	76	77	78	79	80	81	82	83	84	85
Number	1	1	1	1	1	1	3	11	10	22	31	70	78	96	94	149

Table 2.1: Years number of multigrid publications

The first practical results were reported in a pioneering paper by Brandt[44], who published another paper in 1977[45], clearly outlining the main principles and the practical utility of multigrid methods, which drew wide attention and marked the beginning of rapid development. The "multigrid guide" of Brandt[11] has provided to give guidelines for researchers writing their own multigrid programs. In the words of Brandt[11], multigrid methods satisfy the so-called golden rule:

> "The amount of computational work should be proportional to the amount of real physical change in the system. Stalling numerical processes must be wrong."

Multigrid methods can be seen as an acceleration technique for the iterative solution of algebraic systems of equations, or, in their full generality, as a consistent framework enabling the solution of a given problem to be found at a nearly optimal cost. The basic idea is to define a sequence of grids, and to cycle through the different grids in the course of the computations so that the features of the solution are resolved on the most suitable grid. In the finite difference and finite element analysis context, multigrid methods can be considered to be a device for resolving solutions on the coarsest grid possible: low frequency features will be resolved on coarse grids while high frequency features will require finer grids.

The amount of literature pertaining to multigrid methods is vast. Besides the references already outlined others can be underlined. Reference[46] discusses the fundamental ideas of multigrid and defines its vocabulary. References[47, 48] provide a clear introduction to fundamental concepts of multigrid and a useful list of suggested readings. References [49, 50] deal with iterative solvers for system of linear equations and provide a good background to multigrid theory. For important advances in the field of integral equations see[51]. A publication on parabolic multigrid is [52].

More recent literature can be found. Reference [53] is about a multigrid method for numerical simulation of fluid flows in curvilinear grids. For an application of an automatic differentiation method to a 2D Navier-Stokes CFD code for airfoil computations see reference [54]. The multigrid performance for compressible Navier-Stokes equations in low-speed flows can be found in reference [55]. Reference [56] gives some pratical applications of multigrid residual techniques in multigrid. A study on the convergence estimates for multigrid algorithms can be found in reference [57].

2.5.2 Basic Iterative Methods

Basic iterative schemes for the solution of linear systems of equations are marred by their poor asymptotic convergence rates. Some times relaxation methods are local so that iterative solvers remove the high frequency error modes better than they eliminate those at lower frequencies. As the computation progress, the convergence rate is limited by the low-frequency modes in the error. The reason for these phenomena is that low frequencies are usually associated with eigenvalues of the iteration matrix which are close to unity[48]. Suppose that discretization of the partial differential equation to be solved leads to the following linear algebraic system:

$$A\mathbf{y} = \mathbf{b} \tag{2.57}$$

Let the matrix A be split as:

$$A = M - N \tag{2.58}$$

with M non-singular. Then the following iteration method for the solution of Equation (2.57) is called a basic iterative method:

$$\tilde{\mathbf{y}}^{n+1} = S\tilde{\mathbf{y}}^{n} + M^{-1}\mathbf{b}, \qquad S = M^{-1}N \tag{2.59}$$

where $\tilde{\mathbf{y}}^{n+1}$ and $\tilde{\mathbf{y}}^{n}$ are successive approximations of $\mathbf{y}$ and S is the iteration matrix. Defining the error as follows:

$$\mathbf{e}^{n} = \mathbf{y} - \tilde{\mathbf{y}}^{n} \tag{2.60}$$

and assuming that the iterative scheme does not obtain the exact discrete solution, successive error vectors are given by:

$$\mathbf{e}^{n} = S^{n}\mathbf{e}^{0} \tag{2.61}$$

Where $\mathbf{e}^{0}$ is the initial error. $\mathbf{e}^{n}$ can then be expanded on the basis of the k eigenvectors $\mathbf{w}_k$ of S:

$$\mathbf{e}^{n} = \sum_{k=1}^{n} \mathbf{e}_k^{n}\mathbf{w}_k = \sum_{k=1}^{n} \mathbf{e}_k^{0}(\lambda_k)^{n}\mathbf{w}_k \tag{2.62}$$

where λ_k are the n eigenvalues of S. This equation show that the k^{th} mode of the error vector is reduced by a factor of λ_k at each iteration. Hence is damped very slowly, when λ_k is close to unity. Moreover, Equation (2.62) also shows, that after a few iterations, the error is proportional to the largest eingenvector of the system.

If we introduce the vector norm and the associated matrix norm on Equation (2.61) we get the following relationship:

$$\|\mathbf{e}^{n}\| \leq \|S\|^{n}\|\mathbf{e}^{0}\| \tag{2.63}$$

Between the initial error and the error after n iterations. The convergence condition that after M iterations the norm of the error is reduced by a factor of 10^{-d} is approximately satisfied if:

$$[\rho(S)]^{M} \leq 10^{-d} \tag{2.64}$$

where $\rho(S)$, is the spectral radius of the matrix S, defined as:

$$\rho(S) = \max_{i} |\lambda_i(S)|. \tag{2.65}$$

$\rho(S)$ is called the convergence factor and $-\log_{10}(\rho(S))$ is called the convergence rate.

So the iterative method will converge if and only if $\rho(S) < 1$. However, the convergence can be very slow if $\rho(S)$ tends to 1.

Equation (2.62) explains the smoothing property, that many iterative schemes possess. Basically the property consist of eliminating the oscillatory components of the error and leaving the smooth ones. It is a serious limitation of these methods. However, it can be overcome and the remedy is one of the pathways to multigrid.

Figure 2.4: Coarse grid Ω^{2h} and fine grid Ω^h

2.5.3 Two-level Algorithm

In Section 2.5.2 we saw that many standard iterative methods possess the smoothing property. This property makes these methods very effective at eliminating the high-frequency or oscillatory components of the error, while leaving the low-frequency or smooth components relatively unchanged. Hence, the immediate issue is whether these methods can be adapted in some way to make them effective on all error components. The so-called two-level scheme algorithm gives the answer. It involves two grids, a coarse grid Ω^{2h} with grid spacing $2h$ and a fine grid Ω^h with grid spacing h (see Figure 2.4). The basic idea of this algorithm can be outlined as follows. Assume that a particular relaxation scheme has been applied on the fine grid h, until only smooth error components remain. We now ask what these smooth components look like on a coarser grid. Figure 2.5 shows the answer.

A smooth wave with $k = 4$ on grid Ω^h with $N = 12$, as the number of grid points, has been projected directly to the grid Ω^{2h} with 6 grid points. It can be seen that a smooth wave on Ω^h look more oscillatory on Ω^{2h}. To be more precise, note that the grid points of the coarse grid Ω^{2h} are the even-numbered grid points of the fine grid Ω^h. Consider the kth mode of the fine grid evaluated at the even-numbered grid points. If $1 \leq k < N/2$, its components can be written as

$$\omega^h_{k,2j} = \sin(2jk\pi/N) = \sin(jk\pi/(N/2)) = \omega^{2h}_{k,j},\ 1 \leq k < N/2. \tag{2.66}$$

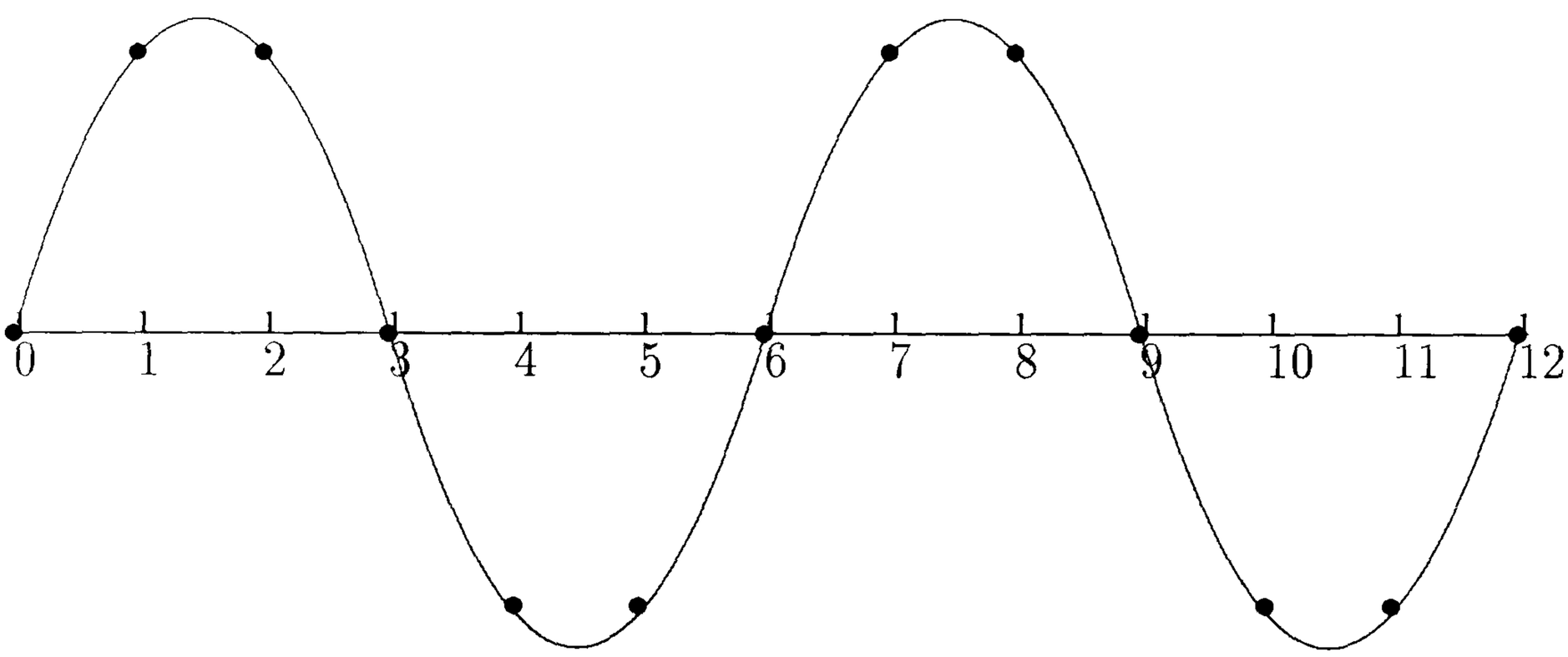
This equation says that the k^{th} mode on Ω^h becomes the k^{th} mode on Ω^{2h}, which means that in passing from the fine grid to the coarse grid, a mode becomes more oscillatory. This suggests that when relaxation begins to stall, signalling the predominance of smooth error modes, it is advisable to move to a coarser grid, on which those smooth error modes appear more oscillatory and relaxation will be more effective. Given this important assessment, we are going to present some necessary definitions to the detailed description of multigrid.

The solution of the problem

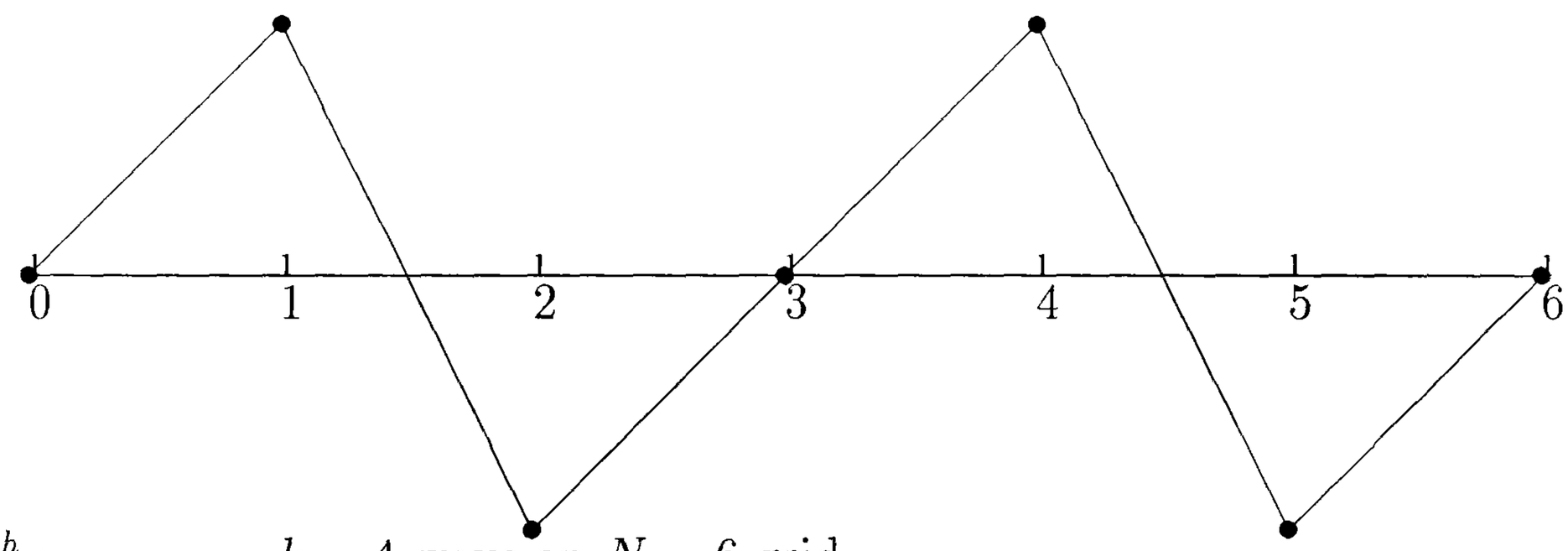
$$A^h\mathbf{u}^h = \mathbf{f}^h \tag{2.67}$$

is sought on the grid Ω^h where h is the meshsize. (2.67) can be solved by the iterative procedure defined by

$$\tilde{\mathbf{u}}^h_{n+1} = P^h\tilde{\mathbf{u}}^h_n + Q^h\mathbf{f}^h \tag{2.68}$$



Ω^h : $k = 4$ wave on $N = 12$ grid



Ω^{2h} : $k = 4$ wave on $N = 6$ grid

Figure 2.5: Wave projections on grid h and grid $2h$

where P^h and Q^h are two matrices and $\tilde{\mathbf{u}}_0^h$ is an initial approximation of $\mathbf{u}^h$. In the context of multigrid method, the iterative procedure is known as the relaxation step. Alternatively, the relaxation procedure is referred to as the smoother, since its main aim is to remove the high frequency components of the error:

$$\mathbf{e}^h = \mathbf{u}^h - \tilde{\mathbf{u}}^h. \tag{2.69}$$

The first stage of a multigrid cycle is to perform some preliminary iterations on the coarse grid to obtain an improved initial guess. Hence, perform ν_1 relaxation sweeps on the initial estimate $\tilde{\mathbf{u}}_0^h$ to obtain an improved approximation $\tilde{\mathbf{u}}_{\nu_1}^h$. Convergence can be greatly accelerated by considering the residual equation:

$$A^h\mathbf{e}^h = \mathbf{r}^h \tag{2.70}$$

where the residual $\mathbf{r}^h$ is defined by:

$$\mathbf{r}^h = \mathbf{f}^h - A\tilde{\mathbf{u}}^h. \tag{2.71}$$

An approximation of $\mathbf{e}^h$ is sought on the coarse grid in order to quickly correct the fine grid approximation. This approach makes sense only if $\mathbf{r}^h$ is smooth so that it can be accurately represented on the coarse grid. If so the restriction stage takes place replacing the problem defined by Equation (2.70) by the coarse grid problem:

$$A^{2h}\mathbf{e}^{2h} = \mathbf{r}^{2h} \tag{2.72}$$

In order to completely define this coarse grid problem, we have to transfer the fine grid residual to the coarse grid and establish the coarse grid operator A^{2h}. To transfer the fine grid residual $\mathbf{r}^h$ to the coarse grid we use a restriction operator I_h^{2h}, so that

$$\mathbf{r}^{2h} = I_h^{2h}\mathbf{r}^h. \tag{2.73}$$

Simple restriction procedures are sufficient to give good results. Two common choices are the injection operator and the full weighting operator, which can be defined by:

$$r_i^{2h} = r_{2i}^h, \qquad 1 \leq i \leq (N/2 - 1)$$

and

$$r_{2h} = \frac{1}{4}(r_{2i-1}^h + r_{2i}^h + r_{2i+1}^h), \qquad 1 \leq i \leq (N/2 - 1)$$

The coarse grid operator A^{2h} is defined by:

$$A^{2h} = I_h^{2h}A^h I_{2h}^h \tag{2.74}$$

where I_h^{2h} is the full weighting restriction operator mentioned above, and I_{2h}^h is the prolongation operator used to transfer the correction to the fine grid (see below). Once an estimate $\tilde{\mathbf{e}}^{2h}$ for $\mathbf{e}^{2h}$ in Equation (2.72) is available, it is transferred back

to the coarse grid. This is the process of prolongation. The approximation $\tilde{\mathbf{u}}^h$ is updated by applying the rule:

$$\tilde{\mathbf{u}}^h \leftarrow \tilde{\mathbf{u}}^h + I_{2h}^h \tilde{\mathbf{e}}^{2h}. \tag{2.75}$$

where I_{2h}^h is an interpolation operator used to transfer the coarse grid correction to the fine grid:

$$\mathbf{u}^h = I_{2h}^h \mathbf{u}^{2h}.$$

In a one-dimensional case, a common choice for this interpolation operator is defined by:

$$\begin{aligned} u_{2j}^h &= u_j^{2h}, & 0 \leq j \leq (N/2-1) \\ u_{2j+1}^h &= \tfrac{1}{2}(u_j^{2h} + u_{j+1}^{2h}), & 0 \leq j \leq (N/2-1) \end{aligned} \tag{2.76}$$

The vector $I_{2h}^h \tilde{\mathbf{e}}^{2h}$ may contain errors due to the interpolation procedure. However these errors are local and therefore mainly consist of high frequency modes which can be efficiently removed by the ν_2 relaxation sweeps that are performed on the fine grid. The procedure outlined above can be repeated until the error is deemed to be sufficiently small, or in practice, the residual be less than a given tolerance.

The different steps of the two-level algorithm can be summarised as follows:

1. Relax on Ω^{2h} ν times using, for example, the weighted Jacobi method.
2. Restrict on Ω^{2h} using the full-weighting operator so that: $\mathbf{f}^{2h} \equiv I_h^{2h}(\mathbf{f}^h - A^h \tilde{\mathbf{u}}^h)$
3. Solve exactly the coarse grid problem: $A^{2h}\tilde{\mathbf{u}}^{2h} = \mathbf{f}^{2h}$
4. Correct the approximation of $\tilde{u}$ on Ω^h: $\tilde{\mathbf{u}}^h \leftarrow \tilde{\mathbf{u}}^h + I_{2h}^h \tilde{\mathbf{u}}^{2h}$.

2.5.4 Multigrid Cycling Strategies

Until this point, the coarse grid problem has been defined. Obviously the coarse grid problem does not differ qualitatively from the fine grid problem, and (for instance) using an even coarser grid with grid spacing double the initial coarsest grid could solve it. This recursive application of the two-level algorithm gives rise to proper multigrid algorithms. The definition of multigrid cycle (or multigrid schedule) specifies the order in which the grids are visited. If the order in which the grids are visited is fixed in advance, we have a fixed cycle (or schedule). Otherwise, if that visiting order depends on intermediate computational results, we have an adaptive cycle (or schedule).

One of the fixed cycles often used is a V-cycle(see Figure 2.6). In a V-cycle the two-level algorithm is applied once in order to solve the coarse grid problem. The cost of the algorithm is very dependent on the amount of relaxation sweeps $\nu_1 + \nu_2$ performed at each visit of a grid. Sufficient relaxation should take place to eliminate high frequencies of the error but too many relaxation sweeps, particularly

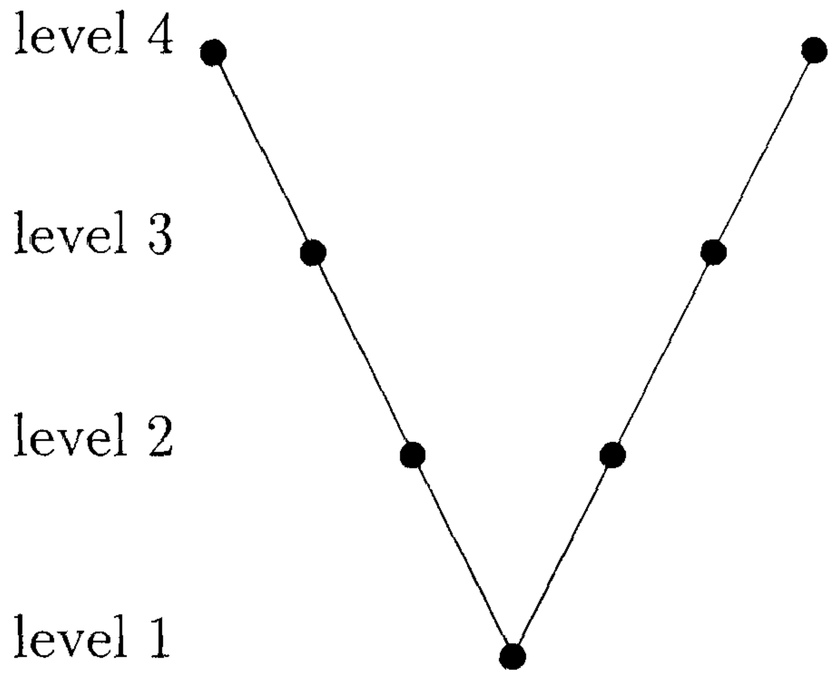


Figure 2.6: Four level multigrid V-Cycle (level 1 coarsest grid)

relaxing the already smooth error vector, may not accelerate converge and greatly penalise the performance. The number of relaxation sweeps to choose is problem dependent, but in practice values in the order of two work well.

A W-cycle is another fixed cycle (see Figure 2.7). It consists of applying twice the two-level algorithm in order to solve the coarse grid problem. Naturally a W-cycle costs more than a V-cycle but the coarse grid correction is of better quality so that fewer cycles are required to solve a problem. Moreover, W-cycles are usually more robust than V-cycles for non-linear problems.

An F-cycle (see Figure 2.8) is another possible choice for a fixed cycle multigrid algorithm. It is quite similar to the W-cycle in the sense that the coarse grid computations are repeated in order to improve the quality of the coarse grid correction. However, it is less costly than the W-cycle.

A variant of the V-cycle is the sawtooth cycle[47](see Figure 2.9). In this cycle the smoothing before coarse grid correction (pre-smoothing) is deleted.

V-cycles and F-cycles are optimal iterative methods in the sense that[58] the computational work required to achieve a fixed accuracy is proportional to the number of discrete unknowns.

Another possible approach to accelerating the converge of iterative methods is based on the concept of nested iteration, which combined with multigrid methods leads to the Full Multigrid Method (FMG)(see, for instance [58], and [48] for a Full Multigrid V-cycle). A Full Multigrid Method (FMG) (see Figure 2.10) achieves the discretization accuracy for an amount of computational work, which is still proportional to the number of discrete unknowns, if the order of the accuracy of the grid transfers is sufficiently high. The key idea is to solve the coarse grid

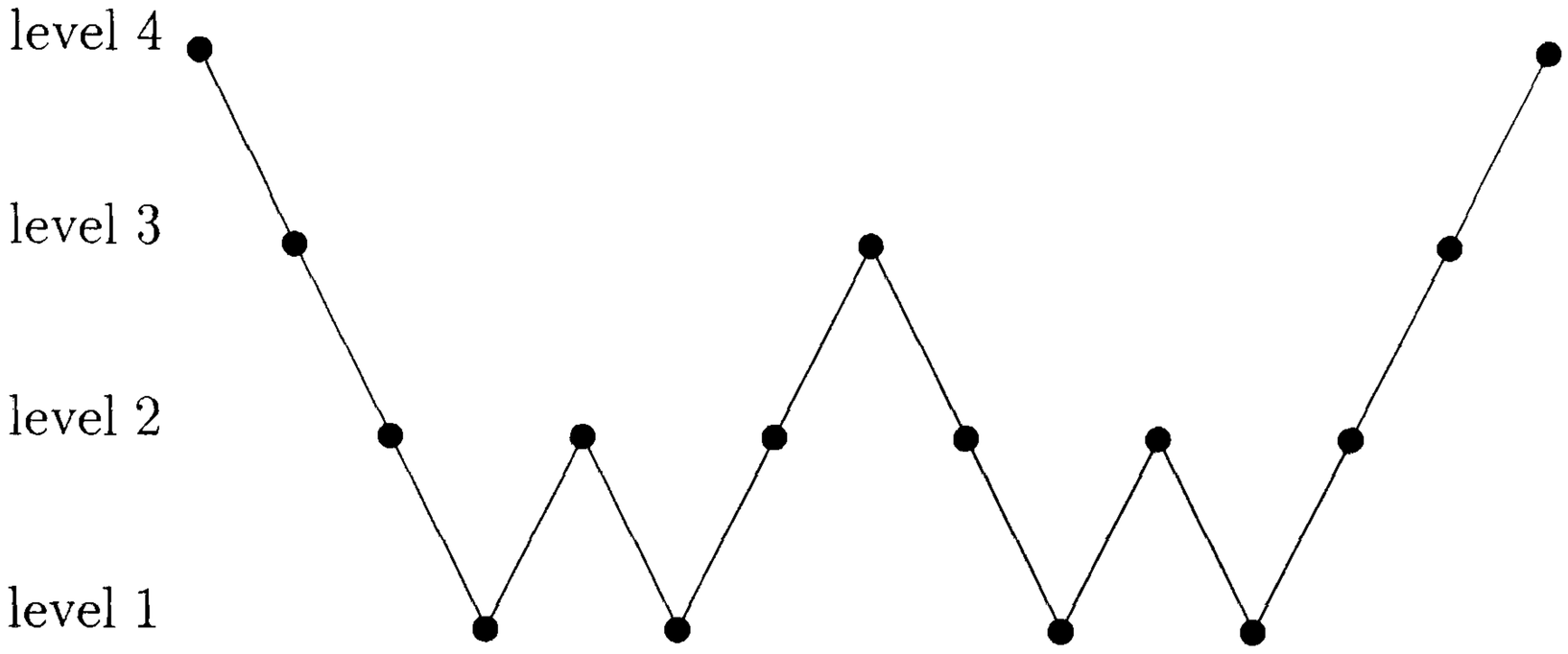


Figure 2.7: Four level multigrid W-Cycle (level 1 coarsest grid)

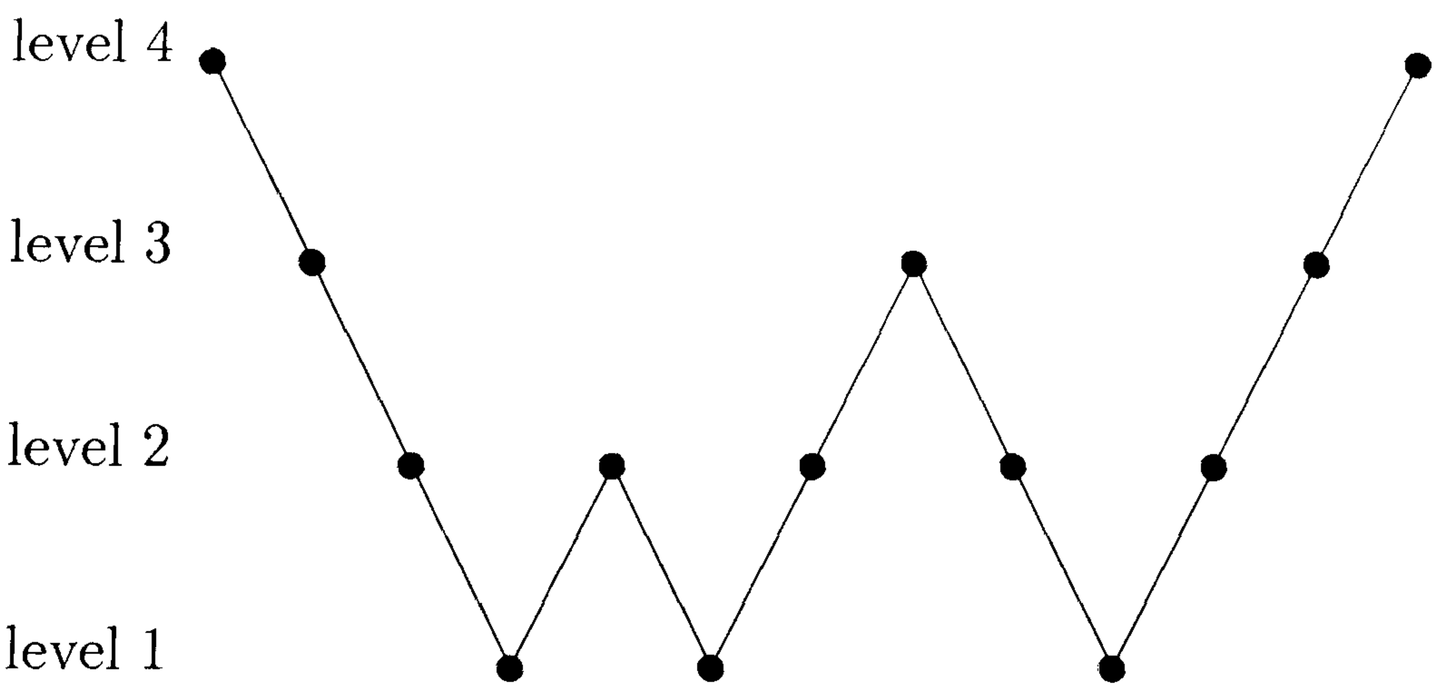


Figure 2.8: Four level multigrid F-Cycle (level 1 coarsest grid)

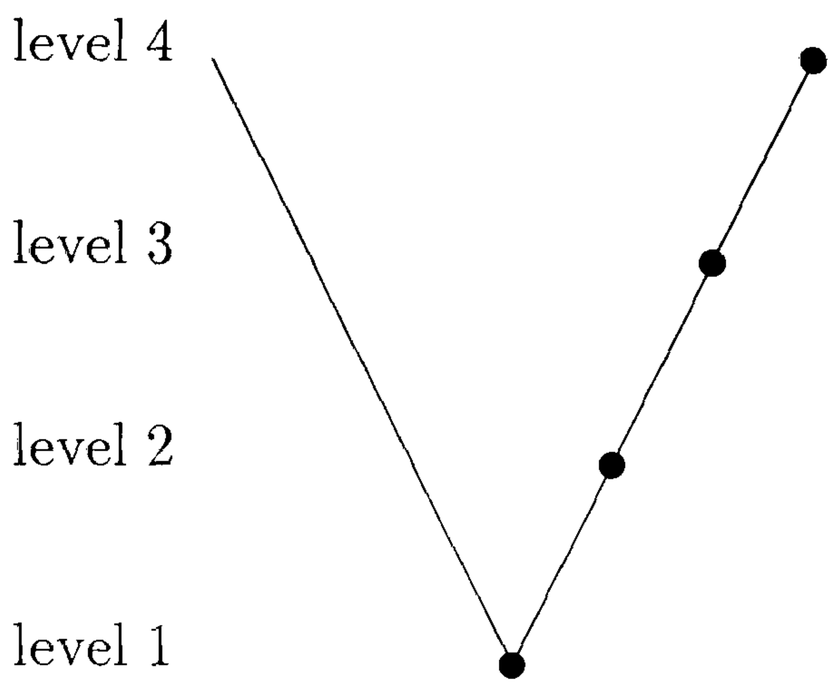


Figure 2.9: Four level multigrid sawtooth-Cycle (level 1 coarsest grid)

problem as accurately as possible before the prolongation to the fine grid takes place. Basically an FMG method starts on the coarsest grid and works itself up to the finest level by repeated application of prolongations and multigrid iterations.

An example of an adaptive cycling strategy, is the following. Suppose we do not carry out a fixed number of multigrid iterations on grid h, but wish to continue to carry out multigrid interactions, until the problem on grid h is solved to within a specified accuracy. Let the accuracy requirement be

$$\|L^h(\mathbf{u}^h) - \mathbf{f}^h\| \leq \epsilon_h = \delta_h \|L^{2h}(\mathbf{u}^{2h}) - \mathbf{f}^{2h}\| \tag{2.77}$$

with $\delta \epsilon (0,1)$ a parameter. At first sight, a more natural definition of ϵ_h would seem to be $\epsilon_h = \delta_h \|\mathbf{f}^h\|$. Since $\mathbf{f}^h$ does not, however, go to zero on convergence, this would lead to skipping of coarse grid correction when $\mathbf{u}^{2h}$ approaches convergence.

2.5.5 Full Approximation Storage Scheme

The Full Approximation Storage Scheme (FAS) variant of multigrid methods is especially designed to solve non-linear problems by extending the correction scheme presented in Section 2.5.3.

Given a sequence of grids Ω^l with $l = 1, ..., k$, the solution u^k to the non-linear system of algebraic equations:

$$\mathcal{L}^k(\mathbf{u}^k) = \mathbf{f}^k \tag{2.78}$$

is sought on the finest grid Ω^k.

In FAS, given an approximation $\tilde{\mathbf{u}}_n^k$ of the solution, we compute a correction $\delta\tilde{\mathbf{u}}^k$ to give the next approximation of the solution:

$$\tilde{\mathbf{u}}_{n+1}^k = \tilde{\mathbf{u}}_n^k + \delta\tilde{\mathbf{u}}^k$$

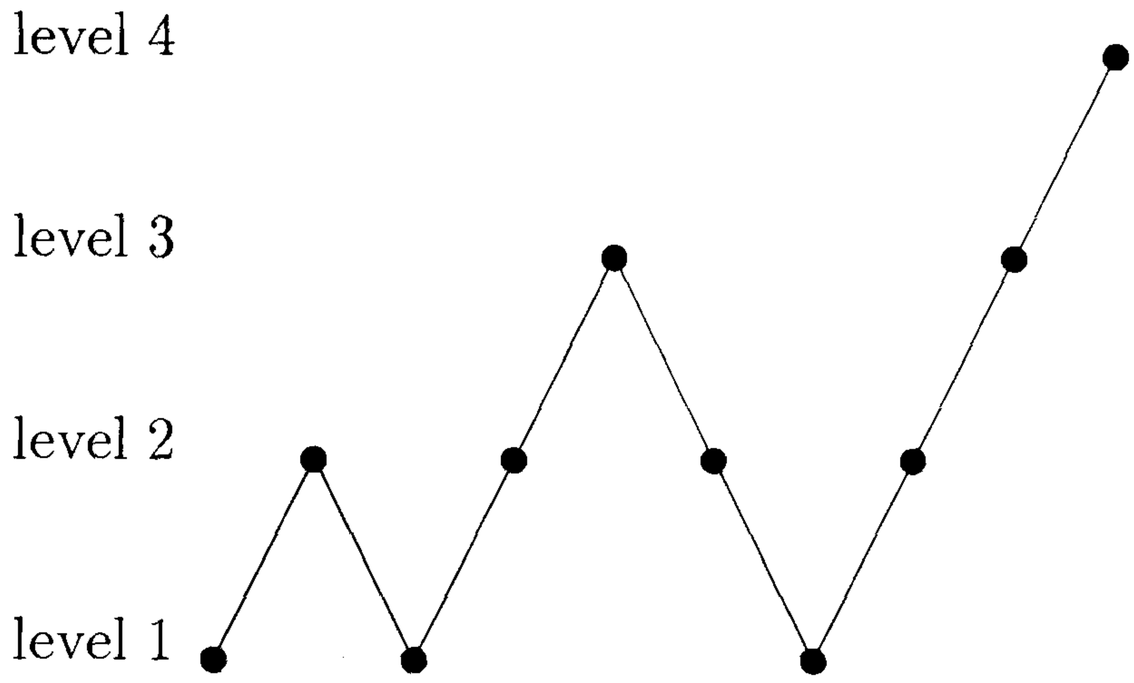


Figure 2.10: Multigrid FMG algorithm

The correction, which is assumed to contain only low frequency modes of the error is then computed on the next coarser grid Ω^{k-1}. This coarse grid correction is then interpolated back to the fine grid using a prolongation operator I_{k-1}^{k}.

The following problem is therefore posed on the immediately coarser grid $k-1$:

$$\mathcal{L}^{k-1}(\tilde{\mathbf{u}}^{k-1}) = \mathbf{f}^{k-1} \tag{2.79}$$

with

$$\mathbf{f}^{k-1} \equiv I_{k}^{k-1}(\mathbf{f}^{k} - \mathcal{L}^{k}(\tilde{\mathbf{u}}^{k})) + \mathcal{L}^{k-1}(I_{k}^{k-1}(\tilde{\mathbf{u}}^{k})) \tag{2.80}$$

Equation (2.80) is obtained by writing a fine grid equation for the correction $\delta\tilde{\mathbf{u}}^{k}$ to the current approximation $\tilde{\mathbf{u}}^{k}$, and transferring it to the coarser grid Ω^{k-1} (see above). After Equation (2.79) has been solved, a coarse grid correction defined as:

$$\delta\tilde{\mathbf{u}}^{k-1} \equiv \tilde{\mathbf{u}}^{k-1} - I_{k}^{k-1}(\tilde{\mathbf{u}}^{k}) \tag{2.81}$$

is computed and transferred back to the fine grid using the transfer operator I_{k-1}^{k}. So far a two grid algorithm has been described. It is obvious that Equation (2.79) can itself be solved more efficiently using the next coarser grid Ω^{k-2}. Some of the cycling strategies discussed in the previous section could also be implemented.

One fundamental difference between the Correction Scheme 2.5.3 and the Full Approximation Storage scheme, is that in the FAS scheme we need to restrict both the residual and the approximation of the fine grid solution to the coarse grid, while in the Correction Scheme only the fine grid residuals need to be interpolated.

The restriction operators in the FAS scheme are not necessarily identical. Let $\hat{I}_{k}^{k-1}$ be the operator working on the solution, and I_{k}^{k-1} the residual operator. Hence

in its general form, the coarse grid correction procedure for FAS can be establish by:

$$\mathcal{L}^{k-1}(\tilde{\mathbf{u}}^{k-1}) = \mathbf{f}^{k-1} \equiv I_k^{k-1}(\mathbf{f}^k - \mathcal{L}^k(\tilde{\mathbf{u}}^k)) + \mathcal{L}^{k-1}(\hat{I}_k^{k-1}\tilde{\mathbf{u}}^k) \tag{2.82}$$

and

$$\tilde{\mathbf{u}}^k \equiv \tilde{\mathbf{u}}^k + I_{k-1}^k(\tilde{\mathbf{u}}^{k-1} - \hat{I}_{k-1}^k\mathbf{u}^k) \tag{2.83}$$

The coarse grid problem represented by Equations (2.82) and (2.83) is consistent: as the fine grid residual $\mathbf{f}^k - \mathcal{L}^k(\tilde{\mathbf{u}}^k)$ tends to zero, $\mathbf{f}^{k-1}$ tends to $\mathcal{L}^{k-1}(\hat{I}_k^{k-1}\tilde{\mathbf{u}}^k)$, the coarse grid solution $\tilde{\mathbf{u}}^{k-1}$ tends to $\hat{I}_k^{k-1}(\tilde{\mathbf{u}}^{k-1})$ and finally, the coarse grid correction tends to zero.

It is often claimed[11] that FAS schemes are directly applicable to non-linear problems and that global linearisations are not only unnecessary but also harmful to its efficiency. This issue takes us to the derivation the formulas for FAS. For a non-linear problem the residual equation written on the fine grid is:

$$\mathcal{L}^k(\mathbf{u}^k) - \mathcal{L}^k(\tilde{\mathbf{u}}^k) = \mathbf{f}^k - \mathcal{L}^k(\tilde{\mathbf{u}}^k) = \mathbf{r}^k \tag{2.84}$$

A correction $\delta\mathbf{u}^k$ is sought such that $\tilde{\mathbf{u}}^k + \delta\mathbf{u}^k \approx \mathbf{u}^k$, therefore the fine grid equation for the correction is:

$$\mathcal{L}^k(\tilde{\mathbf{u}}^k + \delta\mathbf{u}^k) - \mathcal{L}^k(\tilde{\mathbf{u}}^k) = \mathbf{f}^k - \mathcal{L}^k(\tilde{\mathbf{u}}^k) = \mathbf{r}^k \tag{2.85}$$

This equation is then transferred to the coarse grid $k-1$ and so $\mathcal{L}^k, \tilde{\mathbf{u}}^k$ and $\mathbf{r}^k$ are replaced by $\mathcal{L}^{k-1}, \hat{I}_k^{k-1}\tilde{\mathbf{u}}^k$ and $I_k^{k-1}\tilde{\mathbf{r}}^k$ respectively. Establishing

$$\tilde{\mathbf{u}}^{k-1} \equiv \hat{I}_k^{k-1}\tilde{\mathbf{u}}^k + \delta\mathbf{u}^{k-1} \tag{2.86}$$

the FAS scheme is then obtained.

For linear problems FAS and CS are exactly equivalent. However, when $\mathcal{L}$ is non-linear, the discrete operators $\mathcal{L}^{k-1}$ and $\mathcal{L}^k$ are not known. So as the relaxation proceeds good approximations are obtained, but it remains that the coarse grid correction may not constitute a very good fine grid correction, if the problem in-hand is strongly non-linear.

Another important point is that FAS scheme provides an estimate of the truncation error of the algorithm. Introducing the defect τ_k^{k-1} as:

$$\tau_k^{k-1} \equiv \mathcal{L}^{k-1}(\hat{I}_k^{k-1}\tilde{\mathbf{u}}^k) - I_k^{k-1}(\mathcal{L}^k\tilde{\mathbf{u}}^k). \tag{2.87}$$

If the restriction operator I_k^{k-1} is linear, the FAS correction scheme can be rewritten in terms of the defect as:

$$\mathcal{L}^{k-1}\tilde{\mathbf{u}}^{k-1} = \mathbf{f}^{k-1} + \tau_k^{k-1} \tag{2.88}$$

where $\mathbf{f}^{k-1} = I_k^{k-1}\mathbf{f}^k$.

τ_k^{k-1} can also be interpreted as an approximation of the truncation error on the grid $k-1$ where k is the finest grid available.

The truncation error on the grid $k-1$ can be defined as:

$$\tau^{k-1} \equiv \mathcal{L}^{k-1}(\tilde{\mathbf{u}}^{k-1}) - \mathcal{L}(\mathbf{u}) \tag{2.89}$$

So, if the computational error $\mathbf{f}^k - \mathcal{L}(\tilde{\mathbf{u}}^k)$ is small enough so that it can be neglected compared with other errors, then using Equation (2.80), we get:

$$\mathcal{L}^{k-1}(\tilde{\mathbf{u}}^{k-1}) \approx \mathcal{L}^{k-1}(\hat{I}_k^{k-1}\tilde{\mathbf{u}}^k) \tag{2.90}$$

On the grid $k-1$, we can get the following approximation:

$$[\mathcal{L}(\mathbf{u})]^{k-1} \approx I_k^{k-1}(\mathcal{L}^k\tilde{\mathbf{u}}^k). \tag{2.91}$$

This is not applicable to coarser grids k with $1 \leq k < k-1$ because the solution $\tilde{\mathbf{u}}^{k+1}$ converges to a corrected version of the original differential equation. For the purpose of approximating the truncation error of the grid $k-1$, the difference between $\tilde{\mathbf{u}}^k$ and $\mathbf{u}$ is neglected, together with the difference between the non-discrete operator $\mathcal{L}$ and its discretisation $\mathcal{L}^k$ on the grid k. So,

$$\tau_k^{k-1} \approx \tau^{k-1} \tag{2.92}$$

and the defect can be interpreted as an approximation of the truncation error on the grid $k-1$.

2.6 Adaptivity

Recently, several new approaches in computational fluid dynamics have been developed which have the potential of significantly increasing the capabilities of modelling complex flow phenomena and of treating difficult problems in fluid-structure interaction. These new approaches are based on the notions of adaptive methods and smart algorithms. They use instantaneous measures of the quality and other features of the numerical flowfield as a basis for making changes in the structure of the computational grid and of algorithms designed to function on the grid. So, adaptive methods represent an adaptive-control, indeed an optimal-control approach to computational fluid dynamics in which the entire structure of the numerical approximation is dynamically changed to meet certain control objectives. To systematically adapt a grid to the boundary of a body moving through a flow field or to alter grid size, to keep the approximation error within preset bounds, are examples of such methodology.

The philosophy of such techniques is thus to optimise the computational process by using a computational model with a distribution of gridpoints and mesh sizes to produce results of a given quality for the least number of unknowns and at the same time, to satisfy kinematic constraints and boundary conditions associated with moving boundaries, perhaps those of a body moving in the fluid itself. Inherent in such strategies is the use of automatic mesh refinement.

These methodologies can be distinguished as follows[59]:

r- **methods** The redistribution of gridpoints to reduce error or to satisfy kinematical constraints. These approaches include traditional mesh optimisation schemes in which gridpoints migrate in a mesh to equidistribute error.

h- **methods** The mesh size h is an optimisation parameter and is reduced or increased to meet error controls or to aid in the satisfaction of kinematic requirements. These methods include adaptive *h*- refinement/coarsening techniques and mesh embedding techniques.

p- **methods** The spectral order p of the numerical approximation over each gridcell is a parameter that can be increased or decreased to meet various criteria.

combined methods Combinations of the parameters r, h, and p are simultaneously adjusted to control the numerical process.

Although the data structures and complexity of these types of methods are very high, the net result is often excellent. In general, the problem size required (number of unknowns) to achieve a given accuracy is many times less than that of conventional computational methods.

The topic of adaptive techniques for the solution of partial differential equations is an area about which a vast amount of literature is available. See for instance references[39, 60, 61]. For a basic description on unstructured mesh generation and adaptivity see[62]. Reference [63] gives an *h*- adaptive finite element code for solving coupled Navier-Stokes and energy equations on the thermally driven cavity problem. Another recent paper[64], provide a brief survey of error estimation and its application to adaptive solution of complex flows. An application of the Multilevel Adaptive Technique suggested by Brandt[45, 11] on an adaptive parallel multigrid method to solve the two-dimensional Navier-Stokes equations can be found in[65]. An application of an adaptive finite element algorithm for turbulent flows can be found in[66].

In transient simulations, space adaptation will sometimes be complemented by local time stepping where different time steps are taken at different locations on the grid. Normally this is implemented either on physical grounds, if there are regions where the solution varies faster in time than in others, or on numerical grounds to ensure the stability of the integration schemes (see references[67, 68]).

The `transient Pamg` codes used during this project, support space adaptivity. They are prepared to respond to the important issues that automatic and dynamic gridding raises. One of the issues is how the transfer mechanism on the grids interface affect the accuracy of the scheme. Reference[12] shows that for incompressible Navier-Stokes flows it is very important that the fluxes should be conserved across grid interfaces. It is also important that grid refinement should occur only in regions where the solution varies relatively slowly, so that the interpolations performed at grid boundaries do not affect overall accuracy. Another issue is the complexity of the data structures to implement adaptation efficiently. In order to get the maximum benefit of adaptation it is important to minimise the

associated overheads (grid management, solution transfers and error estimations). In `Pamg` and `Pamg-multiphase` the code complexity was reduced by considering structured grids decomposed in quadrants, which are self-similar under refinement. Although `transient Pamg` codes support space adaptivity this technique has not been used in the test cases studied in this project. It was not in the scope of the present thesis.

2.7 Conclusions

In this chapter, some mathematical models for multi-phase flows have been reviewed. Particular attention has been taken to the multi-fluid equations for a two-phase flow. The homogeneous and drift-flux models have been outlined. Segregated and coupled solvers for the Navier-Stokes equations have been discussed.

Numerical methods for solving partial differential equations have been reviewed and illustrated. Particular emphasise has been taken to the stability limits and truncation error.

After a historically introduction to multigrid methods, the basic concepts and definitions have been summarised. These include the two-level scheme and the FAS method. Possible cycling strategies, fixed or adaptive, have been discussed.

Finally a review of the known concepts of adaptivity, has been outlined. Benefits of these types of methods have been discussed. Several examples of the adaptive techniques for the solution of partial differential equations have been outlined.

The main objective of this thesis is twofold. First of all to design an efficient and robust algorithm for the simulation of transient multi-phase flows. This has been done starting from the steady `Pamg` and `Pamg multiphase` solvers. The implementation used relies on an implicit method. This scheme has the advantage of letting the choice of the time step and grid size being more flexible. The variant of multigrid implemented is the FAS method. It has two important features. One is that it is directly applicable to non-linear problems. The other feature is that it provides estimates for the truncation error of the discretization, which can be used for the automatic grid refinement.

The second objective of this thesis is to define a new concept of adaptivity, the D-adaptivity and to implement this new concept in a new solver for the simulation of transient multi-phase flows. This is particularly important when in the presence of complex geometries that can be divided in several regions where the simulation can be treated with different mathematical models. The `D-adaptive solver` relies on the coupling between a one-dimensional solver with a two-dimensional one. However, additional treatment has to be addressed when changing from one region to another. Although the `D-adaptive solver` has been designed for two specific solvers, its main features can be generalised to other solvers.

Chapter 3

Transient PAMG. Governing Equations and Discretization

3.1 Introduction

The physical principle that mass and momentum are conserved during the motion of incompressible fluid flow is the basis for the governing equations, which have been subject of study during this project. These are partial differential equations in an Eulerian framework.

Discretization is the process by which a correspondence between the continuous solutions of the partial differential equations and the discrete algebraic equations solved by numerical codes is established. It is therefore the first stage of the simulation of a physical phenomenon. Finite difference analysis and finite element analysis, for instance, are means of replacing the continuous problem by a discrete one whose solution is close to the solution of the original problem. So it provides an infrastructure for supporting the resolution of the discrete problem. Naturally a good discretization should minimize the truncation error (see Section 2.4.2) introduced by representing the continuous problem on a discrete grid. By other words it should be accurate. However, it is also relevant that the resulting algebraic equations be solved quickly and in a reliably way. Furthermore, properties such monoticity are very useful during the solution process because they permit to reduce the computational cost of the algorithm implemented. Hence, the solver can satisfy the efficiency property defined in Chapter 1.

In this chapter we present the discretization used for solving the governing equations by the two solvers `transient Pamg Single-phase` and `transient Pamg Multi-phase` formulated, in an Eulerian framework, in terms of partial differential equations. Discrete difference equations are obtained by applying the finite volume method to the governing partial differential equations. The discretization is performed, for both single-phase and multi-phase flows, on staggered grids which are well suited to the computation of incompressible flows. The hybrid scheme (see, for instance[69, 70]) is used to derive discrete equations, which can be accurate up to order 2 without generating non-physical oscillations. Both the staggered grids

and hybrid schemes are design choices, which are key factors for the robustness and efficiency of the solution algorithms used in the `transient Pamg Single-phase` and `transient Pamg Multi-phase`.

3.2 Discretization of the Transient Navier-Stokes Equations for an Incompressible Fluid

3.2.1 Governing Equations

The two-dimensional transient flow of an incompressible viscous Newtonian fluid is governed by the following form of the Navier-Stokes Equations (see for instance[2]).

- Conservation of mass

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{3.1}$$

- Conservation of horizontal momentum

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{\partial p}{\partial x} + \nu(\frac{\partial u^2}{\partial x^2} + \frac{\partial u^2}{\partial y^2}) \tag{3.2}$$

- Conservation of vertical momentum

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} = -\frac{\partial p}{\partial y} + \nu(\frac{\partial v^2}{\partial x^2} + \frac{\partial v^2}{\partial y^2}) \tag{3.3}$$

These governing equations are discretized on a staggered grid(see Figure 3.1), where the velocities are defined at the edges of cells and the pressure at the centre. A finite volume approach is followed and a hybrid discretization scheme is used.

Equations (3.1), (3.2) and (3.3) can be rearranged separating the terms in t giving the following set of equations:

$$0 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \tag{3.4}$$

$$\frac{\partial u}{\partial t} = -\frac{\partial u^2}{\partial x} - \frac{\partial uv}{\partial y} - \frac{\partial p}{\partial x} + \nu(\frac{\partial u^2}{\partial x^2} + \frac{\partial u^2}{\partial y^2}) \tag{3.5}$$

$$\frac{\partial v}{\partial t} = -\frac{\partial uv}{\partial x} - \frac{\partial v^2}{\partial y} - \frac{\partial p}{\partial y} + \nu(\frac{\partial v^2}{\partial x^2} + \frac{\partial v^2}{\partial y^2}) \tag{3.6}$$

We are going to introduce some definitions. Let $\mathbf{L}^n$ be the non-linear operator representative of the right hand side of the Equations (3.4), (3.5) and (3.6), at a time step n.

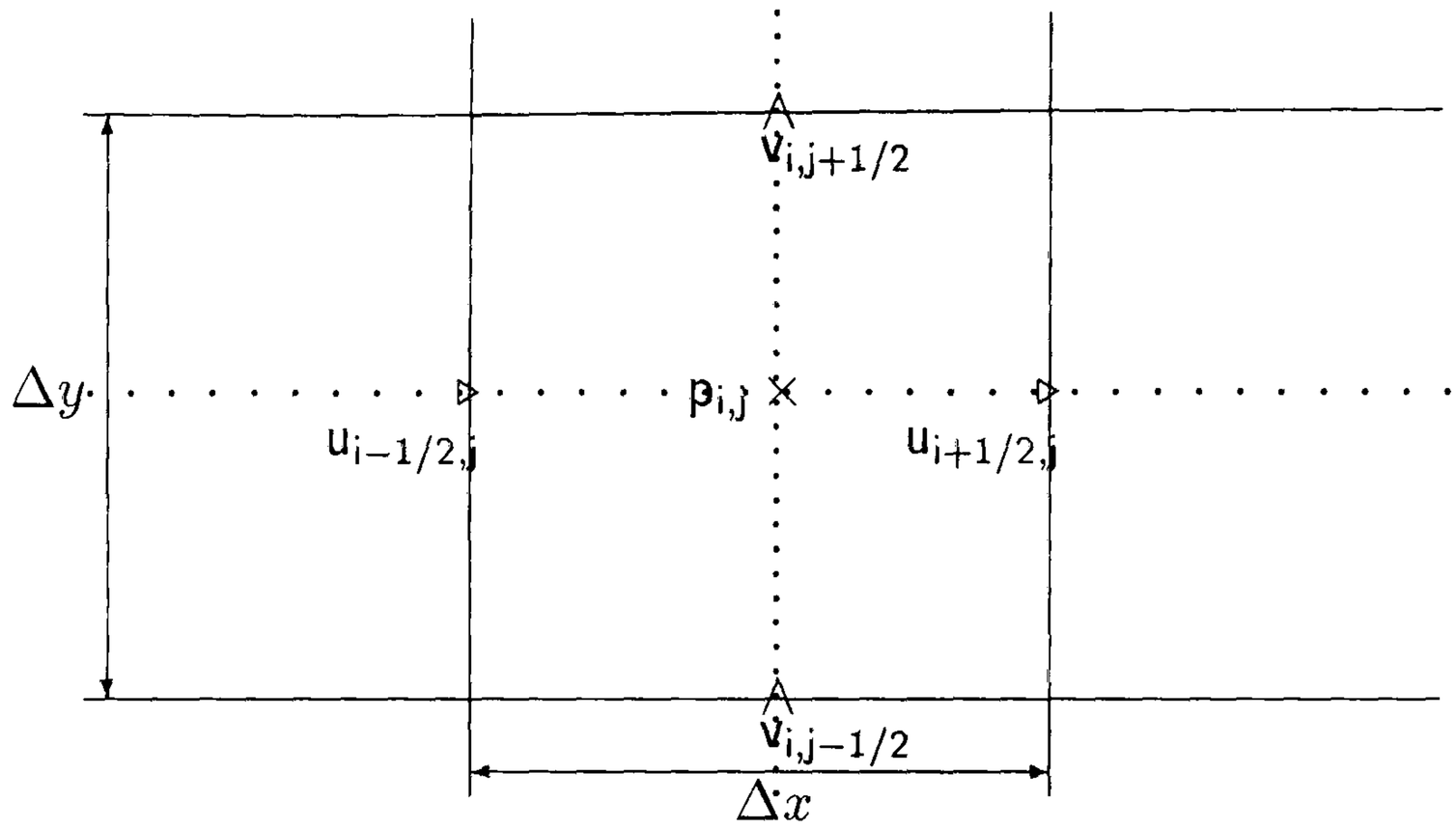


Figure 3.1: Staggered grid used for the discretization of the Navier-Stokes equations

Let $\mathbf{U}^n$ be the vector of components $(u^n_{i-1/2,j}, u^n_{i+1/2,j}, v^n_{i,j-1/2}, v^n_{i,j+1/2}, p^n_{i,j})$ corresponding to the horizontal velocities, vertical velocities and pressure values on a cell (i,j) at time step n.

$\tilde{\mathbf{U}}^n$ the vector of components $(\tilde{u}^n_{i-1/2,j}, \tilde{u}^n_{i+1/2,j}, \tilde{v}^n_{i,j-1/2}, \tilde{v}^n_{i,j+1/2}, \tilde{p}^n_{i,j})$ corresponding to approximations of the horizontal and vertical velocities and pressure on cell (i,j) at time step n.

Hence, Equations (3.4), (3.5) and (3.6) can be approximated using the θ method (see Section 2.4.4) giving:

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} = \theta \mathbf{L}^{n+1}(\tilde{\mathbf{U}}^{n+1}) + (1-\theta)\mathbf{L}^n(\tilde{\mathbf{U}}^n) \tag{3.7}$$

for the time steps n and $n+1$.

However, the $\mathbf{L}^n$ operator for the time step $n+1$ is exactly the operator used in the steady single-phase solver `Pamg`. The discretization equations for the steady Navier-Stokes equations for an incompressible fluid, discretized on a staggered grid by the finite volume method using hybrid schemes, as implemented in `Pamg` solver (see [12]) are:

- Continuity equation

$$\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} = 0 \tag{3.8}$$

- Horizontal momentum equation

$$A^u_C u_{i+1/2,j} = A^u_E u_{i+3/2,j} + A^u_N u_{i+1/2,j+1} + A^u_W u_{i-1/2,j} + A^u_S u_{i+1/2,j-1} - \frac{1}{\Delta x}(p_{i+1,j} - p_{i,j}) \tag{3.9}$$

- Vertical momentum equation

$$A_C^v v_{i,j+1/2} = A_E^v v_{i+1,j+1/2} + A_N^v v_{i,j+3/2} + A_W^v v_{i-1,j+1/2} + A_S^v v_{i,j-1/2} - \frac{1}{\Delta y}(p_{i,j+1} - p_{i,j}) \quad (3.10)$$

with the coefficients defined by

$$A_C^u = A_N^u + A_S^u + A_E^u + A_W^u \quad (3.11)$$

$$A_N^u = \max(|\ C_n^u\ |, D_n^u) - C_n^u \quad (3.12)$$

$$A_S^u = \max(|\ C_s^u\ |, D_s^u) + C_s^u \quad (3.13)$$

$$A_E^u = \max(|\ C_e^u\ |, D_e^u) - C_e^u \quad (3.14)$$

$$A_W^u = \max(|\ C_w^u\ |, D_w^u) + C_w^u \quad (3.15)$$

The Convection coefficients for the horizontal momentum equation are

$$C_n^u = \frac{1}{4\Delta y}(v_{i+1,j+1/2} + v_{i,j+1/2}) \quad (3.16)$$

$$C_s^u = \frac{1}{4\Delta y}(v_{i+1,j-1/2} + v_{i,j-1/2}) \quad (3.17)$$

$$C_e^u = \frac{1}{4\Delta x}(u_{i+3/2,j} + u_{i+1/2,j}) \quad (3.18)$$

$$C_w^u = \frac{1}{4\Delta x}(u_{i+1/2,j} + u_{i-1/2,j}) \quad (3.19)$$

and the Diffusion coefficients for the same equation defined by

$$D_n^u = D_s^u = \frac{\nu}{\Delta y^2} \quad (3.20)$$

$$D_e^u = D_w^u = \frac{\nu}{\Delta x^2} \quad (3.21)$$

Similarly, for the vertical momentum equation we have

$$A_C^v = A_N^v + A_S^v + A_E^v + A_W^v \quad (3.22)$$

$$A_N^v = \max(\mid C_n^v \mid, D_n^v) - C_n^v \tag{3.23}$$

$$A_S^v = \max(\mid C_s^v \mid, D_s^v) + C_s^v \tag{3.24}$$

$$A_E^v = \max(\mid C_e^v \mid, D_e^v) - C_e^v \tag{3.25}$$

$$A_W^v = \max(\mid C_w^v \mid, D_w^v) + C_w^v \tag{3.26}$$

$$C_n^v = \frac{1}{4\Delta y}(v_{i,j+3/2} + v_{i,j+1/2}) \tag{3.27}$$

$$C_s^v = \frac{1}{4\Delta y}(v_{i,j+1/2} + v_{i,j-1/2}) \tag{3.28}$$

$$C_e^v = \frac{1}{4\Delta x}(u_{i+1/2,j+1} + u_{i+1/2,j}) \tag{3.29}$$

$$C_w^v = \frac{1}{4\Delta x}(u_{i-1/2,j+1} + u_{i-1/2,j}) \tag{3.30}$$

$$D_n^v = D_s^v = \frac{\nu}{\Delta y^2} \tag{3.31}$$

$$D_e^v = D_w^v = \frac{\nu}{\Delta x^2} \tag{3.32}$$

Rearranging Equation (3.7) in order of $\mathbf{U}^{n+1}$ and from Equations (3.8), (3.9), (3.10) for the two time steps n and $n+1$, we get the formulas for the horizontal and vertical momentum as

$$\begin{aligned} u_{i+1/2,j}^{n+1} = u_{i+1/2,j}^n + \Delta t\{\theta[&A_N^{u\,(n+1)} u_{i+1/2,j+1}^{n+1} + A_S^{u\,(n+1)} u_{i+1/2,j-1}^{n+1} \\ &+ A_E^{u\,(n+1)} u_{i+3/2,j}^{n+1} + A_W^{u\,(n+1)} u_{i-1/2,j}^{n+1} - A_C^{u\,(n+1)} u_{i+1/2,j}^{n+1} \\ &+ (p_{i,j}^{n+1} - p_{i+1,j}^{n+1})/\Delta x] + (1-\theta)[A_N^{u\,(n)} u_{i+1/2,j+1}^{n} \\ &+ A_S^{u\,(n)} u_{i+1/2,j-1}^{n} + A_E^{u\,(n)} u_{i+3/2,j}^{n} + A_W^{u\,(n)} u_{i-1/2,j}^{n} \\ &- A_C^{u\,(n)} u_{i+1/2,j}^{n} + (p_{i,j}^{n} - p_{i+1,j}^{n})/\Delta x]\} \end{aligned} \tag{3.33}$$

and

$$\begin{aligned} v_{i,j+1/2}^{n+1} = v_{i,j+1/2}^n + \Delta t\{\theta[&A_N^{v\,(n+1)} v_{i,j+3/2}^{n+1} + A_S^{v\,(n+1)} v_{i,j-1/2}^{n+1} \\ &+ A_E^{v\,(n+1)} v_{i+1,j+1/2}^{n+1} + A_W^{v\,(n+1)} v_{i-1,j+1/2}^{n+1} - A_C^{v\,(n+1)} v_{i,j+1/2}^{n+1} \\ &+ (p_{i,j}^{n+1} - p_{i,j+1}^{n+1})/\Delta y] + (1-\theta)[A_N^{v\,(n)} v_{i,j+3/2}^{n} \\ &+ A_S^{v\,(n)} v_{i,j-1/2}^{n} + A_E^{v\,(n)} v_{i+1,j+1/2}^{n} + A_W^{v\,(n)} v_{i-1,j+1/2}^{n} \\ &- A_C^{v\,(n)} v_{i,j+1/2}^{n} + (p_{i,j}^{n} - p_{i,j+1}^{n})/\Delta y]\} \end{aligned} \tag{3.34}$$

In the case of the implicit scheme ($\theta = 1$), we get the approximation formulas for horizontal and vertical momentum as:

$$(1 + \Delta t A_C^{u\,(n+1)}) u_{i+1/2,j}^{n+1} = u_{i+1/2,j}^{n} + \Delta t [A_N^{u\,(n+1)} u_{i+1/2,j+1}^{n+1} + A_S^{u\,(n+1)} u_{i+1/2,j-1}^{n+1} + A_E^{u\,(n+1)} u_{i+3/2,j}^{n+1} + A_W^{u\,(n+1)} u_{i-1/2,j}^{n+1} + (p_{i,j}^{n+1} - p_{i+1,j}^{n+1})/\Delta x] \quad (3.35)$$

$$(1 + \Delta t A_C^{v\,(n+1)}) v_{i,j+1/2}^{n+1} = v_{i,j+1/2}^{n} + \Delta t [A_N^{v\,(n+1)} v_{i,j+3/2}^{n+1} + A_S^{v\,(n+1)} v_{i,j-1/2}^{n+1} + A_E^{v\,(n+1)} v_{i+1,j+3/2}^{n+1} + A_W^{v\,(n+1)} v_{i-1,j+1/2}^{n+1} + (p_{i,j}^{n+1} - p_{i,j+1}^{n+1})/\Delta y] \quad (3.36)$$

The discretization of the continuity equation is identical to the steady state case giving by the Equation (3.8).

3.3 Discretization of the Multi-fluid Equations

In Section 2.2.2 we presented a simplified version of the multi-fluid equations for incompressible fluids. In order to simplify the model, it is assumed that there is no transfer of mass between the phases. Momentum transfer is allowed and is modelled by simple algebraic source terms. Moreover, using the fact that the fluids are incompressible, the momentum equations can be divided by the density of the phase. Under these conditions, and for the simulation of M viscous incompressible fluid flows Equations (2.11) and (2.12) can be written as:

M continuity equations of the form

$$\frac{\partial r_\alpha}{\partial t} + \nabla.(r_\alpha \mathbf{v}_\alpha) = 0. \quad (3.37)$$

M momentum equations of the form

$$\frac{\partial r_\alpha \mathbf{v}_\alpha}{\partial t} + \nabla.(r_\alpha(\mathbf{v}_\alpha \otimes \mathbf{v}_\alpha - \mathbf{T}_\alpha)) = -r_\alpha \frac{1}{\rho_\alpha} \nabla p_\alpha + \frac{1}{\rho_\alpha} \sum_{\beta=1}^{M} C_{\alpha\beta}(\mathbf{v}_\beta - \mathbf{v}_\alpha) \quad (3.38)$$

To close the system we need

M constitutive relationships for the viscous stress

$$\mathbf{T}_\alpha = \nu_\alpha(\nabla \mathbf{v}_\alpha + (\nabla \mathbf{v}_\alpha)^T) \quad (3.39)$$

One closure relationship for the volume fraction

$$\sum_{\alpha=1}^{M} r_\alpha = 1 \tag{3.40}$$

$M-1$ closure relationship for the pressures

$$p_1 = p_2 = ... = p_M \tag{3.41}$$

The algebraic source terms:

$$C_{\alpha\beta}(\mathbf{v}_\beta - \mathbf{v}_\alpha) \tag{3.42}$$

model the momentum transfers. They define the amount of momentum transferred to phase α by phase β. In `transient Pamg multi-phase` the mixture model [21] is adopted for which the momentum transfer coefficients are given by:

$$C_{\alpha\beta} = C_{\beta\alpha} = \begin{cases} \frac{C_D}{d_{\alpha\beta}} \rho_{\alpha\beta} r_\alpha r_\beta \mid \mathbf{v}_\beta - \mathbf{v}_\alpha \mid & \text{if } \alpha \neq \beta \\ 0 & \text{otherwise} \end{cases} \tag{3.43}$$

The drag coefficient C_D and the inter-facial length scale $d_{\alpha\beta}$ are assumed to be constants. $\rho_{\alpha\beta}$, the mixture specific mass, is defined as:

$$\rho_{\alpha\beta} = r_\alpha \rho_\alpha + r_\beta \rho_\beta \tag{3.44}$$

Equations (3.37) and (3.38) can be expressed in terms of the scalar unknowns u_α, v_α, r_α and p_α for each phase.

Writting the continuity equation (3.37) in terms of the scalar variables is quite easy. Simple vector calculus leads to the equation:

$$\frac{\partial r_\alpha}{\partial t} + \frac{\partial r_\alpha u_\alpha}{\partial x} + \frac{\partial r_\alpha v_\alpha}{\partial y} = 0 \tag{3.45}$$

In the momentum equation (3.38) the product $r_\alpha \mathbf{v}_\alpha \otimes \mathbf{v}_\alpha$ gives the following tensor:

$$\begin{pmatrix} r_\alpha u_\alpha u_\alpha & r_\alpha u_\alpha v_\alpha \\ r_\alpha u_\alpha v_\alpha & r_\alpha v_\alpha v_\alpha \end{pmatrix}$$

Hence, we have

$$\nabla.(r_\alpha(\mathbf{v}_\alpha \otimes \mathbf{v}_\alpha)) = \begin{pmatrix} (r_\alpha u_\alpha u_\alpha)_x & (r_\alpha u_\alpha v_\alpha)_y \\ (r_\alpha u_\alpha v_\alpha)_x & (r_\alpha v_\alpha v_\alpha)_y \end{pmatrix}$$

The diffusive term is therefore,

$$\begin{aligned} \nabla.(r_\alpha \nu_\alpha [\nabla \mathbf{v}_\alpha + (\nabla \mathbf{v}_\alpha)^T]) &= \nu_\alpha \frac{\partial}{\partial x_j}(r_\alpha \frac{\partial u_{\alpha i}}{\partial x_j} + r_\alpha \frac{\partial u_{\alpha j}}{\partial x_i}) \\ &= \nu_\alpha \begin{pmatrix} 2(r_\alpha u_{\alpha x})_x + [r_\alpha(u_{\alpha y} + v_{\alpha x})]_y \\ [r_\alpha(v_{\alpha x} + u_{\alpha y})]_x + 2(r_\alpha v_{\alpha y})_y \end{pmatrix} \end{aligned} \tag{3.46}$$

In the case of a single-phase Newtonian flow, Equation (3.46) becomes:

$$\begin{pmatrix} u_{\alpha xx} + u_{\alpha yy} \\ v_{\alpha xx} + v_{\alpha yy} \end{pmatrix} \tag{3.47}$$

for the single-phase α.

This has been established by setting $r_\alpha \equiv 1$ and from

$$\begin{aligned} \nabla.(\nabla \mathbf{v}_\alpha + (\nabla \mathbf{v}_\alpha)^T) &= \frac{\partial}{\partial x_j}(\frac{\partial u_{\alpha i}}{\partial x_j} + \frac{\partial u_{\alpha j}}{\partial x_i}) \\ &= \frac{\partial^2 u_{\alpha i}}{\partial x_j \partial x_j} + \frac{\partial^2 u_{\alpha j}}{\partial x_j \partial x_i} \\ &= \frac{\partial^2 u_{\alpha i}}{\partial x_j \partial x_j} \end{aligned} \tag{3.48}$$

after using the continuity equation to obtain

$$\frac{\partial^2 u_{\alpha j}}{\partial x_j \partial x_i} = 0 \tag{3.49}$$

For multi-phase flows where:

$$\nabla.(r_\alpha[\nabla \mathbf{v}_\alpha + (\nabla \mathbf{v}_\alpha)^T]) = \frac{\partial}{\partial x_j}[r_\alpha(\frac{\partial u_{\alpha i}}{\partial x_j} + \frac{\partial u_{\alpha j}}{\partial x_i})] \tag{3.50}$$

So the corresponding simplification would be:

$$\frac{\partial}{\partial x_{\alpha j}}(r_\alpha \frac{\partial u_{\alpha j}}{\partial x_{\alpha i}}) = 0. \tag{3.51}$$

This equation is not satisfied in general. In fact, after making use of the continuity equation:

$$\frac{\partial r_\alpha u_{\alpha i}}{\partial x_{\alpha i}} = 0. \tag{3.52}$$

we have

$$\frac{\partial}{\partial x_{\alpha j}}(r_\alpha \frac{\partial u_{\alpha j}}{\partial x_{\alpha i}}) = -\frac{\partial}{\partial x_{\alpha j}}(u_{\alpha j} \frac{\partial r_\alpha}{\partial x_{\alpha i}}) \tag{3.53}$$

So the term

$$\frac{\partial}{\partial x_{\alpha j}}(r_\alpha \frac{\partial u_{\alpha j}}{\partial x_{\alpha i}})$$

is equal to the divergence of the second order tensor

$$(u_{\alpha j} \frac{\partial r_\alpha}{\partial x_{\alpha i}}) = \begin{pmatrix} u_\alpha r_{\alpha x} & v_\alpha r_{\alpha y} \\ u_\alpha r_{\alpha x} & v_\alpha r_{\alpha y} \end{pmatrix}$$

which represents the variation of momentum due to the spatial variation of the volume fraction.

If the volume fractions are constant in space this tensor is identically zero. In this case, $\nabla.(r_\alpha \mathbf{T})$ is proportional to:

$$\begin{pmatrix} u_{\alpha xx} + u_{\alpha yy} \\ v_{\alpha xx} + v_{\alpha yy} \end{pmatrix}$$

So, in the scalar variables u_α, v_α, r_α and p_α we have the multi-fluid differential equations for phase α as:

$$\frac{\partial r_\alpha}{\partial t} + \frac{\partial r_\alpha u_\alpha}{\partial x} + \frac{\partial r_\alpha v_\alpha}{\partial y} = 0 \tag{3.54}$$

$$\frac{\partial r_\alpha v_\alpha}{\partial t} + \frac{\partial r_\alpha u_\alpha u_\alpha}{\partial x} + \frac{\partial r_\alpha u_\alpha v_\alpha}{\partial y} - \nu_\alpha [2\frac{\partial (r_\alpha \frac{\partial u_\alpha}{\partial x})}{\partial x} + \frac{\partial (r_\alpha \frac{\partial u_\alpha}{\partial y})}{\partial y} + \frac{\partial (r_\alpha \frac{\partial v_\alpha}{\partial x})}{\partial y}] \tag{3.55}$$
$$= -\frac{r_\alpha}{\rho_\alpha}\frac{\partial p_\alpha}{\partial x} + C_{\alpha\beta}(u_\beta - u_\alpha)$$

$$\frac{\partial r_\alpha v_\alpha}{\partial t} + \frac{\partial r_\alpha u_\alpha v_\alpha}{\partial x} + \frac{\partial r_\alpha v_\alpha v_\alpha}{\partial y} - \nu_\alpha [2\frac{\partial (r_\alpha \frac{\partial v_\alpha}{\partial x})}{\partial x} + \frac{\partial (r_\alpha \frac{\partial v_\alpha}{\partial y})}{\partial y} + \frac{\partial (r_\alpha \frac{\partial u_\alpha}{\partial y})}{\partial y}] \tag{3.56}$$
$$= -\frac{r_\alpha}{\rho_\alpha}\frac{\partial p_\alpha}{\partial y} + C_{\alpha\beta}(v_\beta - v_\alpha)$$

The method used for the discretization of the multi-fluid equations can be considered as an extension of the Navier-Stokes discretization. The staggered grid used for multi-phase flows (see Figure 3.2) is very similar tho that one (see Figure 3.1) used for single-phase flows. The only difference concerns the volume fractions, which like the pressure are cell-centred quantities. So interpolation of volume fractions are necessary for the formulation of the discrete multi-phase continuity equations. Moreover the coefficients C_i and D_i need to be modified to take into account the volume fractions.

The governing equations for multi-phase flows of viscous incompressible fluids within the steady multi-fluid model are[10]:

$$\frac{r_e^c u_{i+1/2,j} - r_w^c u_{i-1/2,j}}{\Delta x} + \frac{r_n^c v_{i,j+1/2} - r_s^c v_{i,j-1/2}}{\Delta y} = 0 \tag{3.57}$$

$$A_C^u u_{i+1/2,j} = A_E^u u_{i+3/2,j} + A_N^u u_{i+1/2,j+1} + A_W^u u_{i-1/2,j} + A_S^u u_{i+1/2,j-1} \tag{3.58}$$
$$+(V_n - V_s) - \frac{r_C^u}{\rho \Delta x}(p_{i+1,j} - p_{i,j})$$
$$+\frac{1}{\rho}[C_{\alpha\beta}^u]_C([u_\beta]_{i+1/2,j} - [u_\alpha]_{i+1/2,j})$$

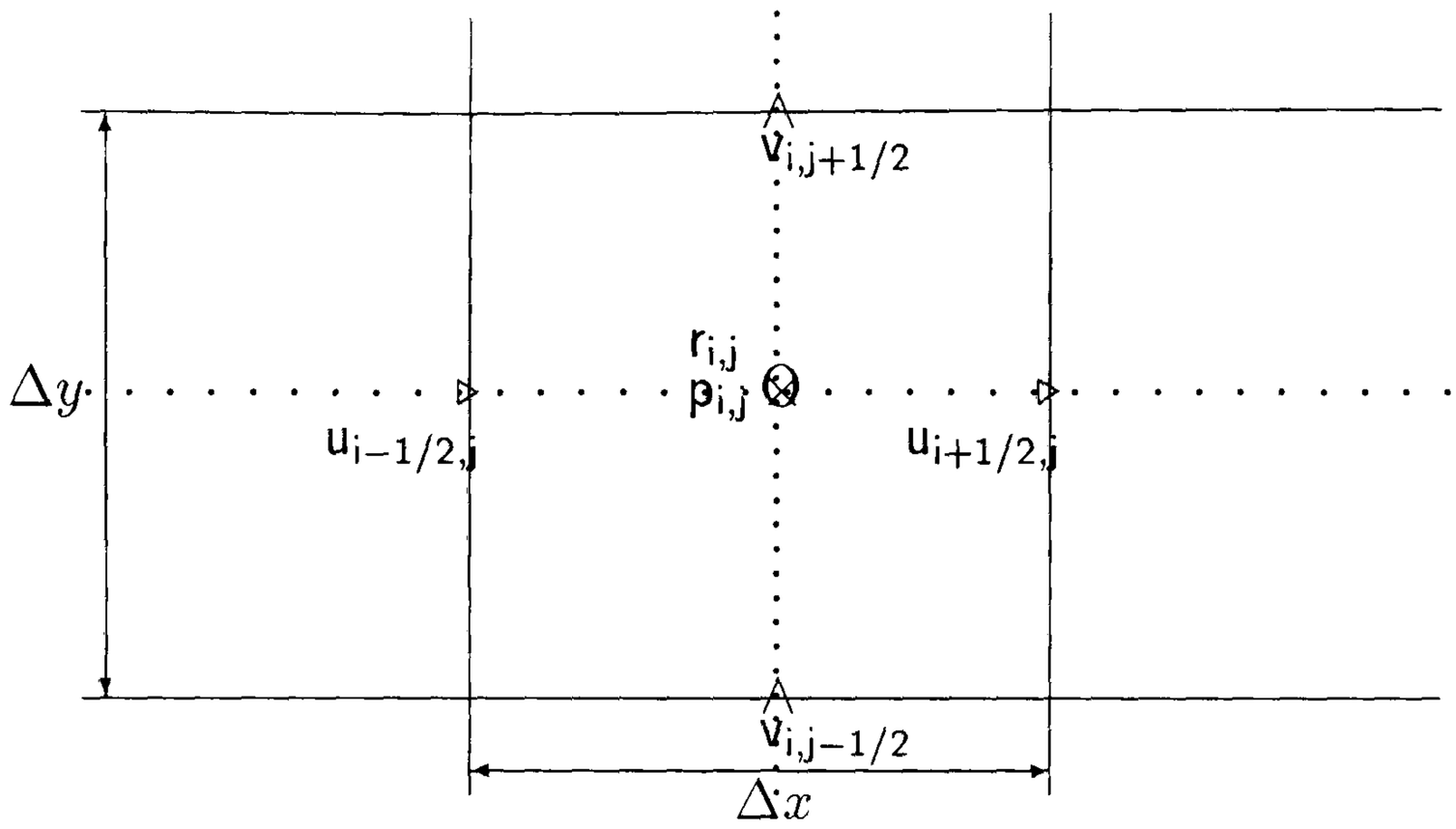


Figure 3.2: Staggered grid used for the discretization of the multi-fluid equations

$$A_C^v v_{i,j+1/2} = A_E^v v_{i+1,j+1/2} + A_N^v v_{i,j+3/2} + A_W^v v_{i-1,j+1/2} + A_S^v v_{i,j-1/2} \qquad (3.59)$$
$$+(U_e - U_w) - \frac{r_C^v}{\rho \Delta y}(p_{i,j+1} - p_{i,j})$$
$$+\frac{1}{\rho}[C_{\alpha\beta}^v]_C([v_\beta]_{i,j+1/2} - [v_\alpha]_{i,j+1/2})$$

with the coefficients defined by

$$A_C^u = A_N^u + A_S^u + A_E^u + A_W^u \qquad (3.60)$$

$$A_N^u = \max(|\, C_n^u \,|, D_n^u) - C_n^u \qquad (3.61)$$

$$A_S^u = \max(|\, C_s^u \,|, D_s^u) + C_s^u \qquad (3.62)$$

$$A_E^u = \max(|\, C_e^u \,|, D_e^u) - C_e^u \qquad (3.63)$$

$$A_W^u = \max(|\, C_w^u \,|, D_w^u) + C_w^u \qquad (3.64)$$

$$C_n^u = \frac{1}{4\Delta y}(v_{i+1,j+1/2} + v_{i,j+1/2}) r_n^u \qquad (3.65)$$

$$C_s^u = \frac{1}{4\Delta y}(v_{i+1,j-1/2} + v_{i,j-1/2}) r_s^u \qquad (3.66)$$

$$C_e^u = \frac{1}{4\Delta x}(u_{i+3/2,j} + u_{i+1/2,j}) r_e^u \qquad (3.67)$$

$$C_w^u = \frac{1}{4\Delta x}(u_{i+1/2,j} + u_{i-1/2,j})r_w^u \tag{3.68}$$

$$D_n^u = \frac{\nu}{\Delta y^2}r_n^u \tag{3.69}$$

$$D_s^u = \frac{\nu}{\Delta y^2}r_s^u \tag{3.70}$$

$$D_e^u = \frac{2\nu}{\Delta x^2}r_e^u \tag{3.71}$$

$$D_w^u = \frac{2\nu}{\Delta x^2}r_w^u \tag{3.72}$$

$$V_n = \frac{1}{\Delta x \Delta y}\nu r_n^u(v_{i+1,j+1/2} - v_{i,j+1/2}) \tag{3.73}$$

$$V_s = \frac{1}{\Delta x \Delta y}\nu r_s^u(v_{i+1,j-1/2} - v_{i,j-1/2}) \tag{3.74}$$

$$[c_{\alpha\beta}^u]_C = \frac{C_D}{d_{\alpha\beta}}([r_\alpha]_C^u\rho_\alpha + [r_\beta]_C^u\rho_\beta)[r_\alpha]_C^u[r_\beta]_C^u \mid [u_\beta]_{i+1/2,j} - [u_\alpha]_{i+1/2,j} \mid \tag{3.75}$$

$$A_C^v = A_N^v + A_S^v + A_E^v + A_W^v \tag{3.76}$$

$$A_N^v = \max(\mid C_n^v \mid, D_n^v) - C_n^v \tag{3.77}$$

$$A_S^v = \max(\mid C_s^v \mid, D_s^v) + C_s^v \tag{3.78}$$

$$A_E^v = \max(\mid C_e^v \mid, D_e^v) - C_e^v \tag{3.79}$$

$$A_W^v = \max(\mid C_w^v \mid, D_w^v) + C_w^v \tag{3.80}$$

$$C_n^v = \frac{1}{4\Delta y}(v_{i,j+3/2} + v_{i,j+1/2})r_n^v \tag{3.81}$$

$$C_s^v = \frac{1}{4\Delta y}(v_{i,j+1/2} + v_{i,j-1/2})r_s^v \tag{3.82}$$

$$C_e^v = \frac{1}{4\Delta x}(u_{i+1/2,j+1} + u_{i+1/2,j})r_e^v \tag{3.83}$$

$$C_w^v = \frac{1}{4\Delta x}(u_{i-1/2,j+1} + u_{i-1/2,j})r_w^v \tag{3.84}$$

$$D_n^v = \frac{2\nu}{\Delta y^2}r_n^v \tag{3.85}$$

$$D_s^v = \frac{2\nu}{\Delta y^2}r_s^v \tag{3.86}$$

$$D_e^v = \frac{\nu}{\Delta x^2}r_e^v \tag{3.87}$$

$$D_w^v = \frac{\nu}{\Delta x^2}r_w^v \tag{3.88}$$

$$U_e = \frac{1}{\Delta x \Delta y}\nu r_e^u(u_{i+1/2,j+1} - u_{i+1/2,j}) \tag{3.89}$$

$$U_w = \frac{1}{\Delta x \Delta y}\nu r_w^u(u_{i-1/2,j+1} - u_{i-1/2,j}) \tag{3.90}$$

$$[c_{\alpha\beta}^v]_C = \frac{C_D}{d_{\alpha\beta}}([r_\alpha]_C^v\rho_\alpha + [r_\beta]_C^v\rho_\beta)[r_\alpha]_C^v[r_\beta]_C^v \mid [v_\beta]_{i,j+1/2} - [v_\alpha]_{i,j+1/2} \mid \tag{3.91}$$

$$r_e^c = 1/2[(1 - sgn(u_{i+1/2,j}))r_{i+1,j} + (sgn(u_{i+1/2,j} - 1))r_{i,j}] \tag{3.92}$$

$$r_w^c = 1/2[(1 - sgn(u_{i-1/2,j}))r_{i,j} + (sgn(u_{i-1/2,j} - 1))r_{i-1,j}] \tag{3.93}$$

$$r_n^c = 1/2[(1 - sgn(v_{i,j+1/2}))r_{i,j+1} + (sgn(v_{i,j+1/2} - 1))r_{i,j}] \tag{3.94}$$

$$r_s^c = 1/2[(1 - sgn(v_{i,j-1/2}))r_{i,j} + (sgn(v_{i,j-1/2} - 1))r_{i,j-1}] \tag{3.95}$$

$$r_e^u = r_{i+1,j} \tag{3.96}$$

$$r_w^u = r_{i,j} \tag{3.97}$$

$$r_n^u = 1/4(r_{i,j} + r_{i+1,j} + r_{i+1,j+1} + r_{i,j+1}) \tag{3.98}$$

$$r_s^u = 1/4(r_{i,j} + r_{i+1,j} + r_{i+1,j-1} + r_{i,j-1}) \tag{3.99}$$

$$r_e^v = 1/4(r_{i,j} + r_{i+1,j} + r_{i+1,j+1} + r_{i,j+1}) \tag{3.100}$$

$$r_w^v = 1/4(r_{i,j} + r_{i,j+1} + r_{i-1,j+1} + r_{i-1,j}) \tag{3.101}$$

$$r_n^v = r_{i,j+1} \tag{3.102}$$

$$r_s^v = r_{i,j} \tag{3.103}$$

$$r_C^u = \frac{1}{2}(r_{i,j} + r_{i+1,j}) \tag{3.104}$$

$$r_C^v = \frac{1}{2}(r_{i,j} + r_{i,j+1}) \tag{3.105}$$

The new time dependent treatment has been done by an implicit method, equivalent to that one implemented in the transient Singe-phase solver. Hence, the discretization equations for the multi-phase incompressible fluid flow are:

- Continuity equation

$$\frac{r_e^c u_{i+1/2,j} - r_w^c u_{i-1/2,j}}{\Delta x} + \frac{r_n^c v_{i,j+1/2} - r_s^c v_{i,j-1/2}}{\Delta y} = 0 \tag{3.106}$$

- Horizontal momentum equation

$$\begin{aligned}(1 + \Delta t A_C^{u\,(n+1)}) u_{i+1/2,j}^{n+1} = u_{i+1/2,j}^n + \Delta t[&A_N^{u\,(n+1)} u_{i+1/2,j+1}^{n+1} + A_S^{u\,(n+1)} u_{i+1/2,j-1}^{n+1} \\ &+ A_E^{u\,(n+1)} u_{i+3/2,j}^{n+1} + A_W^{u\,(n+1)} u_{i-1/2,j}^{n+1} \\ &+ (V_n^{(n+1)} - V_s^{(n+1)}) + r_C^{u\,(n+1)}(p_{i,j}^{n+1} - p_{i+1,j}^{n+1})/\rho\Delta x \\ &+ \frac{1}{\rho}[C_{\alpha\beta}^{u\,(n+1)}]_C([u_\beta{}^{n+1}]_{i+1/2,j} - [u_\alpha{}^{n+1}]_{i+1/2,j})]\end{aligned} \tag{3.107}$$

- Vertical momentum equation

$$\begin{aligned}(1 + \Delta t A_C^{v\,(n+1)}) v_{i,j+1/2}^{n+1} = v_{i,j+1/2}^n + \Delta t[&A_N^{v\,(n+1)} v_{i,j+3/2}^{n+1} + A_S^{v\,(n+1)} v_{i,j-1/2}^{n+1} \\ &+ A_E^{v\,(n+1)} v_{i+1,j+3/2}^{n+1} + A_W^{v\,(n+1)} v_{i-1,j+1/2}^{n+1} \\ &+ (U_e^{(n+1)} - U_w^{(n+1)}) + r_C^{v\,(n+1)}(p_{i,j}^{n+1} - p_{i,j+1}^{n+1})/\rho\Delta y \\ &+ \frac{1}{\rho}[C_{\alpha\beta}^{v\,(n+1)}]_C([v_\beta{}^{n+1}]_{i,j+1/2} - [v_\alpha{}^{n+1}]_{i,j+1/2})]\end{aligned} \tag{3.108}$$

where the coefficients are given by the expressions above, for the steady state incompressible multi-phase case.

3.4 Conclusions

In this Chapter, we have defined the governing equations solved by the two solvers `transient Pamg Single-phase` and `transient Pamg Multi-phase`. The use of staggered grids and hybrid schemes have been used to provide the set of non-linear algebraic equations, which are solved by the steady solvers `Pamg`, and `Pamg multiphase`. After applying an implicit method to the time dependent governing equations the discrete equations have been presented. These like in the steady solvers `Pamg` and `Pamg multiphase` rely on the use of staggered grids and hybrid differencing. The differences between the single-phase discrete equations and the multi-phase discrete equations have been highlighted and discussed.

Chapter 4

Transient PAMG. Single-Phase and Multi-Phase Codes

4.1 Introduction

In this Chapter we are going to present the `transient Pamg Single-phase` and `transient Pamg Multi-phase` solvers which have been developed in the course of this project. Since the steady solvers `Pamg` and `Pamg multiphase` have been the starting point for the transient codes they are also outlined in the first two sections.

The goal is to show the extensions and the new work done to implement the transient solvers from the steady solvers. Moreover, since the `Pamg multiphase` code has been developed from the `Pamg` code (see reference[10]) additional similarities are expected to be found between the `transient Pamg Single-phase` and `transient Pamg Multi-phase`.

Hence, this chapter has been structured in order to separate the steady solvers developed by Thompson and Lezeau, from the transient solvers developed by the author. Furthermore, an effort has been made in order to present the differences between the codes instead of an exhaustive description of each one. If that had been done, these would have been a very high degree of overlap.

The task of extending the `Pamg` to the `transient Pamg Single-phase` required some new features, namely:

- The implementation of an automatic differentiation package to compute the Jacobians used by Newton's method in the solution algorithm.
- The process of guessing the initial condition in order to overcome one of Newton's method limitations-the convergence of the method requires a reasonably close initial guess to the solution.
- The reinitialization process of the initial conditions in order to preserve the algorithmic convergence independent of the number of time steps.

The extension of `Pamg multiphase` to `transient Pamg Multi-phase` did not pose any particular difficult problem. This was due to the successful implementation of the `transient Pamg Single-phase`. Applying the same methodology of development to the `Pamg multiphase` has been followed with no additional difficulty.

4.2 The steady Pamg Single-Phase Algorithm

In this section we describe the various elements of the `Pamg` solution algorithm which has been the starting point for the development of the `transient Pamg Single-phase` algorithm. A complete description of `Pamg` can be found in reference[12]. References[13, 14, 15] also describe some important features of the solver.

4.2.1 Symmetrical Coupled Gauss Seidel

A SCGS (Symmetrical Coupled Gauss Seidel) procedure (see reference [31]) is implemented in the `Pamg` solver. The five-by-five set of equations is solved in a coupled manner. So, two horizontal momentum equations, two vertical momentum equations and one continuity equation are solved simultaneously. This is done on a cell-by-cell basis where the velocities and pressure are corrected locally and simultaneously. The cells are ordered in lexicographic order and during one relaxation sweep, the cells that fill the entire computational domain, are visited in turn. This SCGS procedure is now outlined.

Let

$$A[\mathbf{x}]\mathbf{x} = \mathbf{s} \tag{4.1}$$

where A is a matrix expressing the discrete operator $\mathcal{L}^h$ defined by Equations (3.8), (3.9) and (3.10) written for all cells. The matrix A (which is written as $A[\mathbf{x}]$ to show its dependence on the solution) can be written as $A = D - L - U$ where D, L and U are matrixes respectively diagonal, lower triangular and upper triangular, relative to the lexicographic ordering of the cells. Considering the linear case (matrix A does not depend on $\mathbf{x}$) the standard Gauss Seidel method can be written as:

$$D\hat{\mathbf{x}} = L\mathbf{x}^{(1)} + U\mathbf{x}^{(0)} + \mathbf{s} \tag{4.2}$$

where

$$\mathbf{x}^{(0)}$$

is some initial estimate of the solution vector, and

$$\mathbf{x}^{(1)} = \omega\hat{\mathbf{x}} + (1-\omega)\mathbf{x}^{(0)} \tag{4.3}$$

is the new approximation generated by the Gauss-Seidel procedure. ω is a relaxation parameter. If we introduce the residual vector, defined by

$$\mathbf{r}^{(1)} = \mathbf{s} + L\mathbf{x}^{(1)} + U\mathbf{x}^{(0)} - D\mathbf{x}^{(0)} \tag{4.4}$$

then we get,

$$D(\hat{\mathbf{x}} - \mathbf{x}^{(0)}) = \mathbf{r}^{(1)} \tag{4.5}$$

and the equation to solve is,

$$\frac{1}{\omega} D(\mathbf{x}^{(1)} - \mathbf{x}^{(0)}) = \mathbf{r}^{(1)}. \tag{4.6}$$

Introducing the cell (i, j) and ϕ the group of five variables

$$(u_{i-\tilde{1}/2,j}, u_{i+\tilde{1}/2,j}, v_{i,j\tilde{-}1/2}, v_{i,j\tilde{+}1/2}, p_{\tilde{i},j}) \tag{4.7}$$

we have to solve the following linear system of equations for the correction $\Delta\phi$ to ϕ:

$$\frac{1}{\omega} D_{ij} \Delta\phi = \mathbf{r}_{ij}^{(1)} \tag{4.8}$$

$$\phi^{(1)} = \phi^{(0)} + \Delta\phi \tag{4.9}$$

where D_{ij} is the block of the matrix D which corresponds to the cell (i, j) and $\mathbf{r}_{i,j}^{(1)}$ is the subvector of $\mathbf{r}^{(1)}$ which corresponds to the cell (i, j).

System (4.9) is usually solved by any direct method, for instance by Gaussian elimination.

However, if the problem that we are dealing with is non-linear, the Gauss-Seidel procedure is harder than the linear case outlined above.

One reason for that is that in the non-linear case, the matrices D, L and U have to be evaluated at each step (using the available estimates for the solution). Hence, Equation (4.2) becomes:

$$D[\mathbf{x}^0]\hat{\mathbf{x}} = L[\mathbf{x}^{(0)}]\mathbf{x}^{(1)} + U[\mathbf{x}^{(0)}]\mathbf{x}^{(0)} + \mathbf{s}[\mathbf{x}^{(0)}] = \mathbf{r}^{(1)} \tag{4.10}$$

and the vector equation to solve is:

$$D_{i,j}[\phi^{(0)}]\hat{\phi} = \mathbf{r}_{i,j}^{(1)} \tag{4.11}$$

Although System (4.11) is still a five-by-five system, an iterative procedure is now necessary to solve it. In the `Pamg` algorithm, $\hat{\phi}$ is computed using a single Newton step (see Section 4.2.2 below). If a fully coupled solver is used to compute the solution of System (4.11), this system is solved not on a block-by-block basis but the equations for all cells are relaxed at the same time by Newton's method. This global coupling between all the unknowns carries the computation of Jacobians of very large size.

Hence, it is important to exploit the sparseness of the Jacobian matrices before implementing this approach. However, the algorithm implemented in `Pamg` is a local coupled solver. Local coupled solvers, since they only require local Jacobians,

give an additional contribution to reduce the size of the systems of equations. The evidence that the off-diagonal elements in the full Jacobian are small suggests that the local coupling between the variables dominate over the global coupling. Moreover, Symmetrical Coupled Gauss Seidel algorithms are robust and efficient. On the other hand, multigrid algorithm ensures that global coupling is taken into account and therefore complements the local coupled solver. Furthermore, multigrid methodology (see Section 2.5) ensures the faster resolution of the long wavelengths of the error.

4.2.2 Newton's Method

Newton's method on a general n-dimensional function F can be defined by (see for instance [71]):

$$\mathbf{x}^{k+1} = \mathbf{x}^k - F'(\mathbf{x}^k)^{-1} F(\mathbf{x}^k), \qquad k = 0, 1, ... \tag{4.12}$$

A study of the convergence of Newton's method for a single real equation (1-dimensional function F) can be found in [72].

So, given the vector equation:

$$f(\phi) = \mathbf{s} \tag{4.13}$$

where $f : \Re^5 \longrightarrow \Re^5$ and $\phi^{(0)}$ is an initial approximation of the solution ϕ, Newton's method updates the approximation by the procedure

$$J\Delta\phi = -(f(\phi^{(0)}) - \mathbf{s}) \tag{4.14}$$

where the correction $\Delta\phi$ is given by

$$\phi^{(1)} = \phi^{(0)} + \Delta\phi \tag{4.15}$$

and J is the Jacobian of $\mathbf{f}$ evaluated at $\phi^{(0)}$

$$J_{ij} = \frac{\partial f_i}{\partial \phi_j}(\phi^{(0)}). \tag{4.16}$$

The crucial issue is to obtain a good approximation of J. In the case of `Pamg`, the system is defined by:

$$f(\phi) = D[\phi]\phi \tag{4.17}$$

and the approximation used is $J \approx D[\phi]$

where J is given by:

$$J = D[\phi] = \begin{pmatrix} (A_c^u)_{i-1/2,j} & 0 & 0 & 0 & 1/\Delta x \\ 0 & (A_c^u)_{i+1/2,j} & 0 & 0 & -1/\Delta x \\ 0 & 0 & (A_c^v)_{i,j-1/2} & 0 & 1/\Delta y \\ 0 & 0 & 0 & (A_c^v)_{i,j+1/2} & -1/\Delta y \\ 1/\Delta x & -1/\Delta x & -1/\Delta y & 1/\Delta y & 0 \end{pmatrix} \tag{4.18}$$

In this matrix the entries on positions 12, 21, 34 and 43 which should be

$$(A_e^u)_{i-1/2,j}$$

$$(A_w^u)_{i+1/2,j}$$

$$(A_n^v)_{i,j-1/2}$$

$$(A_s^v)_{i,j+1/2}$$

have been neglected, for a question of simplicity. Experiments have shown that the presence of these entries is not relevant for the global convergence of the iterative scheme. Moreover, the elimination of this entries increases the speed of the computations due to the increase in the sparseness of the matrix J.

System (4.14) can be solved by simple Gaussian elimination. The elimination is done explicitly to exploit the sparseness of J and naturally increase the speed of the computations.

4.2.3 Grid Transfer Operators

We are going to specify the transfer operators chosen for the FAS multigrid algorithm (see section 2.5.5) used in `Pamg`.

For restriction operators the choice has been on second order linear interpolation for the velocities and bilinear interpolation for the pressure. So, for the velocities the coarse grid values are defined by the following relationships (see Figures 4.1 and 4.2):

$$u^{(c)}_{ic+1/2,j} = \frac{1}{2}(u^{(f)}_{if+1/2,jf-1} + u^{(f)}_{if+1/2,jf}) \tag{4.19}$$

$$v^{(c)}_{ic,jc+1/2} = \frac{1}{2}(v^{(f)}_{if-1,jf+1/2} + v^{(f)}_{if,jf+1/2}) \tag{4.20}$$

where the subscripts (c) and (f) refer to coarse and fine grid values respectively. For the pressure we have the expression (see Figure 4.3):

$$p^{(c)}_{ic,jc} = \frac{1}{4}(p^{(f)}_{if-1,jf-1} + p^{(f)}_{if,jf-1} + p^{(f)}_{if-1,jf} + p^{(f)}_{if,jf}) \tag{4.21}$$

This relationship applies for all the interior points only but it is the only restriction formula necessary for the pressure, since the pressure values are defined at the cell centres. At the walls, velocities are null and so no transfer is necessary. Moreover, at the boundary, the formulas for interior points are used and the boundary conditions are applied just after the transfer.

Since the residuals of the momentum equations are defined on cell edges it is therefore possible to use the same operators for the restriction of the velocities. Furthermore, the continuity residuals are defined at the cell centres and so it is

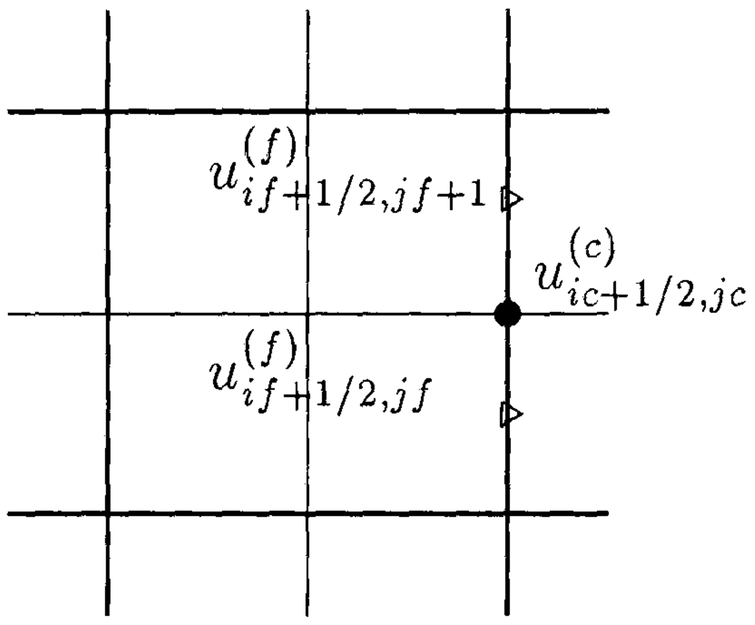


Figure 4.1: Restriction operator for the horizontal velocities

also possible to use the restriction operator for the pressures. Hence, the restriction formulas defined above are generally used for the restriction of the fine grid residuals.

In the FAS algorithm used by `Pamg`, we need to interpolate the corrections rather than the solution values to the fine grid. We obtain the following operators for the prolongation of the velocities, remembering that Δ identifies the corrections:

Horizontal Velocities

$$\Delta u^{(f)}_{if+1/2,jf} = \Delta u^{(c)}_{ic+1/2,jc} + \frac{1}{8}\Delta u^{(c)}_{ic+1/2,jc+1} - \frac{1}{8}\Delta u^{(c)}_{ic+1/2,jc-1} \tag{4.22}$$

$$\Delta u^{(f)}_{if+1/2,jf-1} = \Delta u^{(c)}_{ic+1/2,jc} - \frac{1}{8}\Delta u^{(c)}_{ic+1/2,jc+1} + \frac{1}{8}\Delta u^{(c)}_{ic+1/2,jc-1} \tag{4.23}$$

$$\Delta u^{(f)}_{if-1/2,jf} = \frac{1}{2}(\Delta u^{(c)}_{ic+1/2,jc} + \Delta u^{(c)}_{ic-1/2,jc}) + \frac{1}{16}(\Delta u^{(c)}_{ic+1/2,jc+1} + \Delta u^{(c)}_{ic-1/2,jc+1}) - \frac{1}{16}(\Delta u^{(c)}_{ic+1/2,jc-1} + \Delta u^{(c)}_{ic-1/2,jc-1}) \tag{4.24}$$

$$\Delta u^{(f)}_{if-1/2,jf-1} = \frac{1}{2}(\Delta u^{(c)}_{ic+1/2,jc} + \Delta u^{(c)}_{ic-1/2,jc}) - \frac{1}{16}(\Delta u^{(c)}_{ic+1/2,jc+1} + \Delta u^{(c)}_{ic-1/2,jc+1}) + \frac{1}{16}(\Delta u^{(c)}_{ic+1/2,jc-1} + \Delta u^{(c)}_{ic-1/2,jc-1}) \tag{4.25}$$

Vertical Velocities

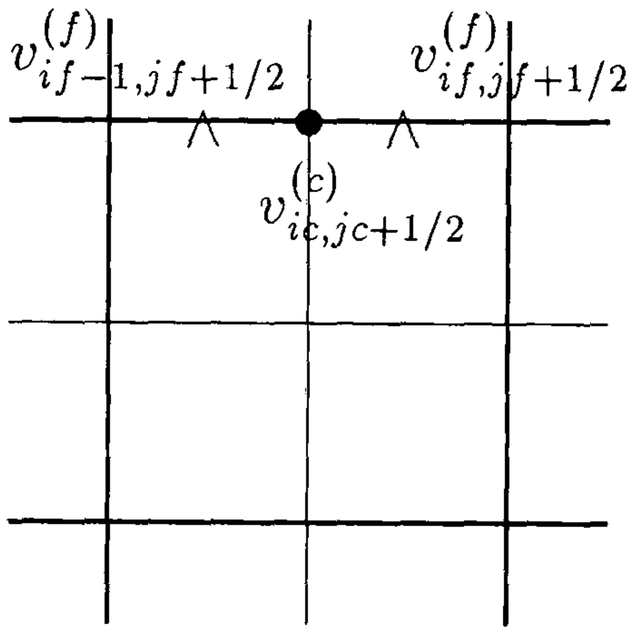


Figure 4.2: Restriction operator for the vertical velocities

$$\Delta v^{(f)}_{if,jf+1/2} = \Delta v^{(c)}_{ic,jc+1/2} + \frac{1}{8}\Delta v^{(c)}_{ic+1,jc+1/2} - \frac{1}{8}\Delta v^{(c)}_{ic-1,jc+1/2} \tag{4.26}$$

$$\Delta v^{(f)}_{if-1,jf+1/2} = \Delta v^{(c)}_{ic,jc+1/2} - \frac{1}{8}\Delta v^{(c)}_{ic+1,jc+1/2} + \frac{1}{8}\Delta v^{(c)}_{ic-1,jc+1/2} \tag{4.27}$$

$$\Delta v^{(f)}_{if,jf-1/2} = \frac{1}{2}(\Delta v^{(c)}_{ic,jc-1/2} + \Delta v^{(c)}_{ic,jc+1/2}) + \frac{1}{16}(\Delta v^{(c)}_{ic+1,jc+1/2} + \Delta v^{(c)}_{ic+1,jc-1/2}) - \frac{1}{16}(\Delta v^{(c)}_{ic-1,jc+1/2} + \Delta v^{(c)}_{ic-1,jc-1/2}) \tag{4.28}$$

$$\Delta v^{(f)}_{if-1,jf-1/2} = \frac{1}{2}(\Delta v^{(c)}_{ic,jc+1/2} + \Delta v^{(c)}_{ic,jc-1/2}) - \frac{1}{16}(\Delta v^{(c)}_{ic+1,jc+1/2} + \Delta v^{(c)}_{ic+1,jc-1/2}) + \frac{1}{16}(\Delta v^{(c)}_{ic-1,jc+1/2} + \Delta v^{(c)}_{ic-1,jc-1/2}) \tag{4.29}$$

Near the walls these formula are slightly modified.
For the prolongation of the pressure corrections we get the formulas:

$$\Delta p^{(f)}_{if,jf} = \Delta p^{(c)}_{ic,jc} \tag{4.30}$$

$$\Delta p^{(f)}_{if,jf-1} = \Delta p^{(c)}_{ic,jc} \tag{4.31}$$

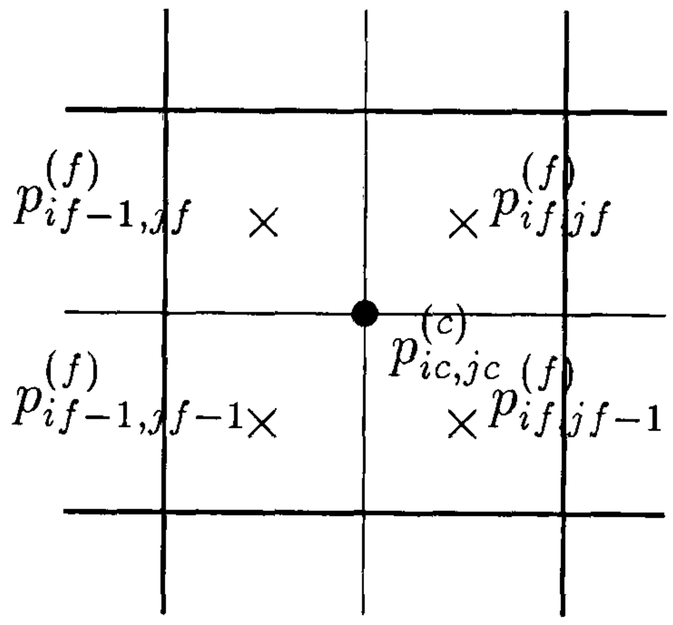


Figure 4.3: Restriction operator for the pressure

$$\Delta p^{(f)}_{if-1,jf} = \Delta p^{(c)}_{ic,jc} \tag{4.32}$$

$$\Delta p^{(f)}_{if-1,jf-1} = \Delta p^{(c)}_{ic,jc} \tag{4.33}$$

which are accurate to first order. However, we can adopt the second order formulae:

$$\Delta p^{(f)}_{if,jf} = \frac{1}{16}(9\Delta p^{(c)}_{ic,jc} + 3\Delta p^{(c)}_{ic+1,jc} + 3\Delta p^{(c)}_{ic,jc+1} + \Delta p^{(c)}_{ic+1,jc+1}) \tag{4.34}$$

$$\Delta p^{(f)}_{if,jf-1} = \frac{1}{16}(9\Delta p^{(c)}_{ic,jc} + 3\Delta p^{(c)}_{ic+1,jc} + 3\Delta p^{(c)}_{ic,jc-1} + \Delta p^{(c)}_{ic+1,jc-1}) \tag{4.35}$$

$$\Delta p^{(f)}_{if-1,jf} = \frac{1}{16}(9\Delta p^{(c)}_{ic,jc} + 3\Delta p^{(c)}_{ic-1,jc} + 3\Delta p^{(c)}_{ic,jc+1} + \Delta p^{(c)}_{ic-1,jc+1}) \tag{4.36}$$

$$\Delta p^{(f)}_{if-1,jf-1} = \frac{1}{16}(9\Delta p^{(c)}_{ic,jc} + 3\Delta p^{(c)}_{ic-1,jc} + 3\Delta p^{(c)}_{ic,jc-1} + \Delta p^{(c)}_{ic-1,jc-1}). \tag{4.37}$$

In all these formulas we have $if = 2ic$ and $jf = 2jc$.

4.2.4 Adaptivity

The FAS scheme is very suitable to adaptive computations (see Section 2.5.5). So, `Pamg` which relies on the FAS multigrid algorithm naturally supports adaptive computations. Adaptivity in `Pamg` necessitates the design of the prolongation and restriction operators, which conserve mass across grid interfaces. Another important

feature of Pamg is that adaptive grids are generated automatically as the solution proceeds. This evidences the self-adaptive power of the solution algorithm. This is supported on a refinement algorithm which can be described as follows. Given a grid at level k, solve the equations to obtain an estimate of the truncation errors for that problem. Based on the estimate of the truncation errors, refine the grid at level k in regions of high truncation errors to obtain the composite grid at level $k+1$. If $k+1$ is already the highest level for which the solution is sought, then solve the equations for the required tolerance. Otherwise, solve them to obtain an estimate of the truncation errors, in order to proceed the refinement up to level $k+2$.

Introducing the concept of patch, which is basically a $n+m$ group of neighbouring cells, in Pamg, each patch contains four-by-four cells. Each patch is thought of as containing four quadrants of equal size two-by-two cells. So, a patch at level $k+1$ exactly matches a quadrant at level k and a locally refined grid at level $k+1$ is simply obtained by overlaying certain quadrants at level k with level $k+1$ patches. Hence, a uniform grid at level $k+1$ is simply obtained by refining all the quadrants at level k. This implementation based on the similarity of refined and non-refined objects facilitated the design of the computer code. Moreover, it gives a strong argument in favour of structured grids.

In Pamg, any non-refined quadrant at level k which satisfies:

$$[\tau_{k+1}^k] \geq \left(\frac{1}{Nvis}\sum_{Nvis}[\tau_{k+1}^k]^\gamma\right)^{1/\gamma} \tag{4.38}$$

is refined, where $[\tau_{k+1}^k]$ is the average L_2 norm of the defect (computed on grid $k-1$) over the quadrant. $Nvis$ is the number of level k quadrants which are yet unrefined, and are therefore eligible for refinement, on all levels. γ is a parameter which controls the extent of the refinement.

Besides γ, adaptive computations are defined by two parameters: k_u the level of the finest uniform grid ($k_u \geq 2$) and k_f the finest level at which the solution is sought.

To get the maximum benefit from the refinement algorithm it is important that the approximate solutions obtained with the uniform grid at level k_u be sufficiently representative of the exact solution to provide a reliable estimate of the truncation error. Normally, if k_u is small, more gains are available from adaptivity. However, if the starting uniform grid is too coarse, it is possible that some features of the flow disappear on that k_u grid, and consequently may be difficult to recapture them.

When composite grids are used, there are refined and unrefined quadrants at any level. A composite grid up to level k can contain visible quadrant on all levels from grid k_u inclusive. In order to obtain the composite grid at level $k+1$, all visible quadrants-not just those at level k-are eligible for refinement. Once the final grid is obtained, the flow solution is accepted when the average residual, on the finest level, is lower than the required tolerance. With this condition we can ensure that the average residual on all levels is lower than the required tolerance. Although

Pamg supports adaptive computations it has never been used in the course of this project. This was not in the scope of the present thesis.

4.3 The steady Pamg Multi-phase Algorithm

In this section, we are going to describe some features of the Pamg multiphase solution algorithm for the simulation of steady incompressible multi-phase fluid flows. A complete description of this solver, developed by Lezeau, can be found in [10]. This multi-phase solver, quite naturally, has the same structure as the single-phase solver Pamg described in Section 4.2. However some important differences have to be outlined. Besides the number of equations and unknowns, which is larger in the multi-phase solver, the main difference is in the quasi-Newton solver. The multi-phase regime demanded a more careful treatment of the solution algorithm in order to get convergence. Restriction and prolongation operators also had to be extended to the volume fractions, but this does not pose any new conceptual problem.

4.3.1 Equations. Automatic Differentiation

Pamg multiphase like Pamg is also based on the Symmetrical Coupled Gauss Seidel procedure (SCGS). Hence, only the discrete equations solved for each cell has to be defined. Considering a single computational cell (i, j), for each phase, we write a continuity equation, two horizontal momentum equations and two vertical momentum equations. These equations have already be outlined in Section 3.3.

The problem is closed by

$$\sum_{\alpha=1}^{M} (r_\alpha)_{i,j} = 1 \tag{4.39}$$

where M is the number of phases.

So, the procedure requires that the local non-linear system of algebraic equations

$$f(\phi) = 0 \tag{4.40}$$

be solved by Newton's method. ϕ is the vector representing, for a two-phase fluid flow, the twelve unknowns defined on the (i, j) computational cell. Hence, the expression of ϕ is

$$\begin{aligned}\phi = [&(u_1)_{i-1/2,j}, (u_1)_{i+1/2,j}, (v_1)_{i,j-1/2}, (v_1)_{i,j+1/2}, (p_1)_{i,j}, (r_1)_{i,j},\\ &(u_2)_{i-1/2,j}, (u_2)_{i+1/2,j}, (v_2)_{i,j-1/2}, (v_2)_{i,j+1/2}, (p_2)_{i,j}, (r_2)_{i,j}]\end{aligned} \tag{4.41}$$

Newton's method applied to Equation (4.40) gives us:

$$J\Delta\phi = -f(\phi^{(n)}) \tag{4.42}$$

where J is the Jacobian of the matrix of f on the cell (i, j) and $\Delta\phi$ is the Newton correction.

Hence, the new approximation of the solution can be updated by:

$$\phi^{(n+1)} = \phi^{(n)} + \Delta\phi \tag{4.43}$$

The key issue for this method, as we already saw in Section 4.2.2, is to obtain an expression for the Jacobian J. In the case of multi-phase flows such expressions for the Jacobian need to be more accurate than in the single-phase case. For instance, an extension of the approximation of the Jacobian used in `Pamg` has been inefficient to guarantee the convergence of the method. This is due to the increased non-linearity of the solutions. Automatic differentiation is a powerful tool to obtain accurate expressions for the Jacobian (see for instance [18]). Its basic principle is to compute the derivative of an expression implemented in a computer program, by applying the chain rule of differentiation. Hence, one advantage of automatic differentiation is that it can compute the derivatives of any mathematical function expressed as a sequence of elemental operations and functions for which the differentiation rules are known. So, it is not necessary to deduce, in a traditional way, an expression of the derivatives in terms of the independent variables. Automatic differentiation provides values for the derivatives corresponding to one set of independent variables. Another advantage of automatic differentiation is that it can be easily incorporated in any developed software. Moreover it may be applied whenever Jacobians of complicated functions are sought.

Some automatic differentiation packages are implemented as pre-processors. These packages (see for instance[19]) implement a program to compute the derivatives of a given mathematical function in a particular programming language. This language usually is Fortran[19, 20] but it can be C[73] too.

In `Pamg multiphase` the package AD01 [20] is used. It is a collection of routines, which computes the derivatives of an expression at run-time stage of the computation. Although the higher cost in terms of performance due to the calculation of the derivatives at run-time stage it has important advantages for software development. It reduces the probability of implementing inconsistent errors between the residual expressions and the Jacobian. Moreover, changing the discretization options can be easily done with minimal code writing.

4.3.2 Line Searching

Newton's method is not globally convergent. One of its limitations is that if the initial guess is not close enough to the solution, the method may not convergence due to the size of the corrective steps, which are taken. This is a consequence of the non-linearity of the equations to be solved. If the degree of non-linearity of the equation increases, the domain of validity of the Newton linearisation may be reduced and the large correction steps which are usually taken when the residual

is large, become more and more inaccurate. So, the scheme will fail to reach convergence to the desired solution.

A simple procedure that can make Newton's method globally convergent is line searching (see reference [74]). Line searching relies on the following methodology: if the Newton correction leads to an increase in the norm (normally the Euclidean norm) of the residual, then the correction is progressively reduced until it results in a reduction of the residual.

Line searching can be summarised as follows for the Equation (4.42). Given the correction $\Delta\phi$, the new approximation of the solution is given by:

$$\phi^{(n+1)} = \phi^{(n)} + \lambda\Delta\phi \tag{4.44}$$

where the scaling factor λ is chosen so that the correction reduces the Euclidean norm of f. Hence

$$\|\phi^{(n+1)}\|_2 < \|\phi^{(n)}\|_2 \tag{4.45}$$

The issue is how to choose λ efficiently. It is possible to find λ in order to minimize $\|\phi^{(n+1)}\|_2$. However any correction that reduces the residual is acceptable.

In `Pamg multiphase` the first try is to apply the full Newton correction in order to benefit from the quadratic convergence rate near the solution. If the Newton step does not reduce the residual, the correction step is progressively reduced until a satisfactory value of λ is found. Following [74] we can describe the process of choosing λ as follows. Let g be a function of λ given by:

$$g(\lambda) = f(\phi^{(n)} + \lambda\Delta\phi) \tag{4.46}$$

So we have

$$g^{'}(\lambda) = \nabla f.\Delta\phi \tag{4.47}$$

If we need to backtrack, then we model g with the most current information we have and choose λ to minimize the model. We start calculating $g(0)$ and $g^{'}(0)$. The first step is always the Newton step, which corresponds to $\lambda = 1$. If this step is not acceptable, so does not reduce the residual, we have to compute $g(1)$ as well. With the three values $g(0)$, $g(1)$ and $g^{'}(0)$ we can therefore model $g(\lambda)$ as a quadratic:

$$g(\lambda) \approx [g(1) - g(0) - g^{'}(0)]\lambda^2 + g^{'}(0)\lambda + g(0) \tag{4.48}$$

Taking the derivative of this quadratic, we find that it is a minimum when

$$\lambda = -\frac{g^{'}(0)}{2[g(1) - g(0) - g^{'}(0)]} \tag{4.49}$$

Since the Newton step failed, it can be shown that $\lambda \leq 1/2$. However, we need to guard against too small values of λ. We set $\lambda_{min} = 0.1$. On subsequent backtracks,

we model g as a cubic in λ, using the previous values $g(\lambda_1)$ and the most recent value

$$g(\lambda_2) : g(\lambda) = a\lambda^3 + b\lambda^2 + g^{'}(0)\lambda + g(0) \tag{4.50}$$

Requiring this expression to give the correct values of g at λ_1 and λ_2 gives two equations that can be solved for the coefficients a and b:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\lambda_1 - \lambda_2} \begin{bmatrix} 1/\lambda_1^2 & -1/\lambda_2^2 \\ -\lambda_2/\lambda_1^2 & \lambda_1/\lambda_2^2 \end{bmatrix} \cdot \begin{bmatrix} g(\lambda_1) - g^{'}(0)\lambda_1 - g(0) \\ g(\lambda_2) - g^{'}(0)\lambda_2 - g(0) \end{bmatrix} \tag{4.51}$$

The minimum of the Cubic (4.50) is at

$$\lambda = \frac{-b + \sqrt{b^2 - 3ag'(0)}}{3a} \tag{4.52}$$

We enforce that λ must lie between $\lambda_{max} = 0.5\lambda_1$ and $\lambda_{min} = 0.1\lambda_1$.

The algorithm outlined above can be considered to be a form of under-relaxation. The value of the correction step depends on how good the correction is. It is possible to relax the correction after line-searching. The way of implementing this is very simple. Introducing a relaxation parameter λ_r the new approximation $\phi^{(n+1)}$ is given by:

$$\phi^{(n+1)} = \phi^{(n)} + \lambda_r LS(\Delta\phi) \tag{4.53}$$

where $LS(\Delta\phi)$ is the correction resulting from the line-searching algorithm.

4.3.3 Transfer Operators. Boundary Conditions. Adaptivity

The transfer operators used in `Pamg multiphase` are based on the extension of the operators used by `Pamg`. The only new quantities that require some special attention are the volume fractions. Since they are cell-centred, their treatment is very similar to the treatment of the pressures.

The multi-phase operators, which restrict the continuity and momentum residuals, are identical to the single-phase operators, since they are defined at identical locations of cells on grids. Naturally, the number of phases gives us the number of times that the residuals need to be transferred.

Since the volume fractions are cell-centred quantities, it is therefore possible to use the same interpolation operators as for the pressures. So, for the restriction we use the formula given for the restriction of the pressures, Equation (4.21). The prolongation of the volume fractions can be defined by the same formulas given for the prolongation of the pressures, namely Equations (4.30) to (4.33) (for first order accuracy) or (4.34) to (4.37) (for second order accuracy).

An important point that must be referred in the prolongation stage of the volume fractions is that they can become non-physical, meaning that their values

do not satisfy the closure law (Equation (4.39)). Naturally, this is more likely when the volume fractions are close to 0 or 1, or when the residuals are still very high and consequently originating large corrections.

`Pamg multiphase` deals with this physical problem, testing the fine grid values after the correction process and reset them to the physical admissible values, using the following algorithm:

$$\begin{aligned} &if \quad r_{ij}{}^{(f)} > 1, \quad set \quad r_{ij}{}^{(f)} = 1 - \epsilon \\ &if \quad r_{ij}{}^{(f)} < 0, \quad set \quad r_{ij}{}^{(f)} = \epsilon \end{aligned} \tag{4.54}$$

where ϵ is a small number (10^{-6}), just added to guarantee that the correction systems are not singular.

Alternatively, `Pamg multiphase` can use another strategy, which is basically not to implement the volume fraction corrections if their sum with the old volume fractions lie outside the interval $[0, 1]$.

The boundary conditions implemented in the `Pamg multiphase` are the following:

- At solid walls: $u = 0$, $v = 0$ and $\partial_n r = 0$, where ∂_n is the derivative in the $n$ormal direction.
- At outlets $v = 0$, $\partial_n u = 0$ and $\partial_n r = 0$.

So, boundary conditions for the volume fractions are applied at solid walls and outlets.

`Pamg multiphase` like `Pamg` also supports adaptivity. The refinement algorithm is not substantially different from the single-phase version. In the multi-phase case, defects are computed over quadrants for each phase separately, using the same procedure as in the single-phase case (Equation (4.38)).

The defect $[\tau_{k+1}^k]$ of the multi-phase regime is then defined as the arithmetic average over the defects of each phase.

4.4 The transient Pamg Single-Phase Algorithm

In this section we are going to present the `transient Pamg Single-phase` derived from the steady `Pamg` single-phase described in Section 4.2. The focus will be on the new developments that have to be taken on the transient solver.

4.4.1 Solution Algorithm

The solution algorithm implemented in `transient Pamg Single-phase` is, quite naturally the Full Approximation Storage Scheme (FAS). The strategy of correcting the velocities and pressures is still based on a Symmetrical Coupled Gauss Seidel procedure. The velocities and the pressure are corrected in a coupled manner on a cell by cell basis, for each time step based on an implicit method.

FAS applied to this new non-linear time dependent problem, is outlined above.

Let $\mathcal{L}^k$ be the discrete non-linear operator defined for the steady state case (see Section 4.2.1).

The transient problem solved by `transient Pamg single-phase` can be defined by:

$$\frac{\mathbf{u}_k^{n+1} - \mathbf{u}_k^n}{\Delta t} = \mathcal{L}^k(\mathbf{u}_k^{n+1}) \tag{4.55}$$

where Δt is the time step, $\mathbf{u}_k^n$ and $\mathbf{u}_k^{n+1}$ the unknowns in grid k for two consecutive time steps n and $n+1$.

Rearranging Equation (4.55) we get

$$(I - \Delta t\mathcal{L}^k)(\mathbf{u}_k^{n+1}) = \mathbf{u}_k^n \tag{4.56}$$

where I is the identity matrix.

The residual equation on the fine grid (grid k) is given by:

$$(I - \Delta t\mathcal{L}^k)(\mathbf{u}_k^{n+1}) - (I - \Delta t\mathcal{L}^k)(\tilde{\mathbf{u}_k}^{n+1}) = \mathbf{u}_k^n - (I - \Delta t\mathcal{L}^k)(\tilde{\mathbf{u}_k}^{n+1}) = \mathbf{r}_k \tag{4.57}$$

where $\tilde{\mathbf{u}_k}^{n+1}$ is an approximation of the unknown $\mathbf{u}_k^{n+1}$.

A correction $\Delta\tilde{u_k}^{n+1}$ is sought such that

$$\tilde{\mathbf{u}_k}^{n+1} + \Delta\tilde{\mathbf{u}_k}^{n+1} \approx \mathbf{u}_k^{n+1} \tag{4.58}$$

So, the fine grid equation for the correction is

$$(I - \Delta t\mathcal{L}^k)(\tilde{\mathbf{u}_k}^{n+1} + \Delta\tilde{\mathbf{u}_k}^{n+1}) - (I - \Delta t\mathcal{L}^k)(\tilde{\mathbf{u}_k}^{n+1}) = \mathbf{u}_k^n - (I - \Delta t\mathcal{L}^k)(\tilde{\mathbf{u}_k}^{n+1}) = \mathbf{r}_k \tag{4.59}$$

Transferring to the coarse grid (grid $k-1$) we get

$$(I - \Delta t\mathcal{L}^{k-1})(\hat{I}_k^{k-1}(\tilde{\mathbf{u}_k}^{n+1}) + \Delta\tilde{\mathbf{u}_{k-1}}^{n+1}) - (I - \Delta t\mathcal{L}^{k-1})(\hat{I}_k^{k-1}\tilde{\mathbf{u}_k}^{n+1}) = I_k^{k-1}\tilde{\mathbf{r}_k} \tag{4.60}$$

where $\hat{I}_k^{k-1}$ is the restriction operator of the solution and I_k^{k-1} is the restriction operator of the residual computed in grid k, $\tilde{\mathbf{r}_k}$.

Although these two restriction operators can be different, in `transient Pamg Single-phase` we use exactly the same operators for both the solution and the residual equations.

Equation (4.60) is equivalent to

$$(I - \Delta t\mathcal{L}^{k-1})(\hat{I}_k^{k-1}(\tilde{\mathbf{u}_k}^{n+1}) + \Delta\tilde{\mathbf{u}_{k-1}}^{n+1}) = (I - \Delta t\mathcal{L}^{k-1})(\hat{I}_k^{k-1}\tilde{\mathbf{u}_k}^{n+1}) + I_k^{k-1}\tilde{\mathbf{r}_k} \tag{4.61}$$

So, setting

$$\tilde{\mathbf{u}_{k-1}}^{n+1} = \hat{I}_k^{k-1}(\tilde{\mathbf{u}_k}^{n+1}) + \Delta\tilde{\mathbf{u}_{k-1}}^{n+1} \tag{4.62}$$

we get the FAS algorithm.

The implementation done on `transient Pamg` works with fixed time steps. The code could be easily modified to work with varying time steps. However, since an implicit method has been used, the convergence does not depend on the value of the stability limit (see Section 2.4).

4.4.2 Automatic Differentiation. `Adifor`

One substantial difference that `transient Pamg Single-phase` has from the original steady `Pamg` concerns the implementation of Newton's method for relaxing and solving the Navier-Stokes equations.

In the beginning of the solver development the Gauss Seidel scheme used in `Pamg` did not show efficiency, particularly when dealing with the smooth components of the error. Strong evidence of a costly number of iterations to get convergence suggested this. This was due to the strongest non-linearity of the time dependent Navier-Stokes equations.

The Jacobian matrix derived for the transient case applying exactly the same technique has in the steady case (see Section 4.2) is:

$$J = D(\phi) = \begin{pmatrix} A_{1,1} & 0 & 0 & 0 & \Delta t/\Delta x \\ 0 & A_{2,2} & 0 & 0 & -\Delta t/\Delta x \\ 0 & 0 & A_{3,3} & 0 & \Delta t/\Delta y \\ 0 & 0 & 0 & A_{4,4} & -\Delta t/\Delta y \\ 1/\Delta x & -1/\Delta x & -1/\Delta y & 1/\Delta y & 0 \end{pmatrix} \tag{4.63}$$

where

$$A_{1,1} = (1 + \Delta t)(A_c^u)_{i-1/2,j} \tag{4.64}$$

$$A_{2,2} = (1 + \Delta t)(A_c^u)_{i+1/2,j} \tag{4.65}$$

$$A_{3,3} = (1 + \Delta t)(A_c^v)_{i,j-1/2} \tag{4.66}$$

$$A_{4,4} = (1 + \Delta t)(A_c^v)_{i,j+1/2} \tag{4.67}$$

Newton's method applied to this new transient problem is therefore:

$$J(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^n) = -R(\tilde{\mathbf{u}}^n) \tag{4.68}$$

where $(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^n)$ is the correction and $R(\tilde{\mathbf{u}}^k)$ the column vector constructed with the five residuals of the approximations of the five unknowns

$$(u_{i-\tilde{1}/2,j}, u_{i+\tilde{1}/2,j}, v_{i,j\tilde{-1}/2}, v_{i,j\tilde{+1}/2}, p_{\tilde{i},j}).$$

However, the implementation of Newton's method has to be much more careful. In particular, the approximation methodology used in the steady `Pamg` for the Jacobian, has proved to be inefficient. Hence, the approximation of the Jacobian given by Equation (4.63) has to be improved. Naturally the first attempt has been to include the neglected entries on the positions 12, 21, 34 and 43 of the Jacobian (discussed in Section 4.2.2). However no significant improvement has been observed.

The automatic differentiation technique has been the obvious choice for computing more accurate expressions for the Jacobians.

The automatic differentiation is based on the application of the chain rule[18]

$$\frac{\partial}{\partial t} f(g(t)) \mid_{t=t_0} = (\frac{\partial}{\partial s} f(s) \mid_{s=g(t_0)})(\frac{\partial}{\partial t} g(t) \mid_{t=t_0}) \tag{4.69}$$

over and over again to the composition of a sequence of elementary operations such as additions, multiplications and elementary functions such as sin, cos, max.

The package `Adifor` (see for instance[18, 19]) has been chosen for computing the required Jacobians.

One of the great advantages of `Adifor` is that only requires that the user supply the source code and indicate the variables that correspond to the independent and dependent variables. So, it is not necessary to calculate the expressions for the derivatives. This can be a very hard task, particularly in the presence of expressions involving several dependent and independent variables. Moreover, some particular cases which require special treatment, for instance the treatment in the walls, can increase the number of expressions which derivatives have to be calculated.

We demonstrate the use of `Adifor`, with a very simple program[19] shown in Table 4.1. On Table 4.2 the code produced by `Adifor` to compute the derivative of *squareroot* at a specified value of x is demonstrated. Finally, Table 4.3 represents the derivative code *driver* for the very simple example of computing the *squareroot* at a specified value of x and the derivative of the *squareroot* at the same value x.

In `transient Pamg Single-phase` the use of `Adifor` has the main objective of computing the derivatives of the $R(\tilde{\mathbf{u}}^n)$ column vector constructed with the five residuals of the approximations of the five unknowns

$$(u_{i-\tilde{1}/2,j}, u_{i+\tilde{1}/2,j}, v_{i,j\tilde{-1}/2}, v_{i,j\tilde{+1}/2}, p_{\tilde{i},j}).$$

Hence, Newton's method applied to the time dependent equations solved by `transient Pamg Single-phase` can be defined by:

$$(\frac{\partial R(\tilde{\mathbf{u}}^n)}{\partial \tilde{\mathbf{u}}^n})(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^n) = -R(\tilde{\mathbf{u}}^n) \tag{4.70}$$

where $\frac{\partial R(\tilde{\mathbf{u}}^k)}{\partial \tilde{\mathbf{u}}^k}$ is the expression for the Jacobian computed by the automatic differentiation package.

```
program main
real x, y
read *, x
call squareroot(x,y)
print *, y
end

subroutine squareroot(x,y)
real x, y
y = sqrt(x)
end
```

Table 4.1: Program main to compute the square root of x

4.4.3 Initial Conditions. Reinitialization Process

As we discussed in Section 4.3.2 one of the requirements for the convergence of Newton's method is a reasonably close initial guess. Since reasonably and close are subjective terms it is up to the user to determine what makes an initial guess a good initial guess. To minimize this problem, reference[75] says that Venkatakrishnan among others added a term to the Jacobian matrix diagonal. This term essentially turns Newton's method into a backward Euler[4] formulation.

The new equation to be solved is then:

$$(\frac{1}{\Delta\tau}(I) + \frac{\partial R(\tilde{\mathbf{u}}^n)}{\partial \tilde{\mathbf{u}}^n})(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^n) = -R(\tilde{\mathbf{u}}^n) \tag{4.71}$$

where I is the identity matrix, and $\Delta\tau$ the pseudo-time step.

Equation (4.71) becomes Equation (4.70) when $\Delta\tau \to \infty$. Normally a small initial value of $\Delta\tau$ is chosen. When a new value of $\tilde{\mathbf{u}}^n$ is obtained, the residual $R(\tilde{\mathbf{u}}^n)$ and its norm can be calculated. Choosing the Euclidean norm for this computation, a new value of $\Delta\tau$ is obtained by the expression:

$$\Delta\tau_{new} = \Delta\tau \frac{\| R^n(\mathbf{u}^k) \|_2}{\| R^{n+1}(\tilde{\mathbf{u}^k}) \|_2} \tag{4.72}$$

where n and $n+1$ represent two consecutive iterations, and $\| \cdot \|_2$ the Euclidean norm.

Equation (4.72) shows that as the norm of the residual decreases the pseudo-time step increases. Furthermore, the quality of the approximation $\tilde{\mathbf{u}^k}$ is reflected in the new pseudo-time step $\Delta\tau_{new}$. This technique permits considerably more latitude in choosing an initial guess. However, the computation time can increase substantially.

```
subroutine q-squareroot(g-p-, x, g-x, ldg-x, y, g-y, ldg-y)
real x, y
integer g-pmax-
parameter (g-pmax-=1)
integer g-i-, g-p-, ldg-y, ldg-x
real r1-p, r2-v, g-y(ldg-y), g-x(ldg-x)
if (g-p- .gt. g-pmax-) then
   print *, 'Parameter g-p- is greater than g-pmax-'
   stop
endif
r2-v=sqrt(x)
if ( x .gt. 0.0e0) then
   r1-p=1.0e0 / (2.0e0 * r2-v)
else
   call ehufSV (9, x, r2-v, r1-p, 'g-squareroot.f', 32)
endif
do g-i-=1, g-p-
    g-y(g-i-) = r1-p * g-x(g-i-)
enddo
y=r2-v
end
```

Table 4.2: Derivative code of the subroutine *squareroot*, generated by `Adifor`

So, if a large number of iterations are necessary to increase the time step, Newton's method quadratic convergence is not achieved as quickly, and therefore its power is deeply reduced. Moreover, since direct inversion of the global Jacobian matrix is computationally expensive, the longer it takes to achieve quadratic convergence the more expensive the method becomes.

In `transient Pamg Single-phase` the methodology outlined above has been implemented. However, the test cases studied did not demand such approximation treatment of the initial guess. This is due to the effective robustness of the algorithm implemented.

In the time dependent problems that have been the object of study in this thesis, evidence has been found that after a certain number of time steps iterations, the new solution to look after in the next time step iterative process is the same solution computed in the last time step iterative process. In other words, after a certain number of time step iterations the transient simulation of the fluid flow reach the steady state case, as one would expect.

However, another important issue that we have to deal in the `transient Pamg Single-phase` has to be with what we call the reinitialization process of the initial

```
program driver
real x,y
real g-x(1), g-y(1)

real *, x
g-x=1.0
call g-squareroot(1, x, g-x, 1, y, g-y, 1)

print *, y
print *, g-y
end
```

Table 4.3: Derivative code *driver* for the main program

condition. This basically relies on the necessity of restarting the initial condition after a certain number of iterations, due to the propagation of the error.

This reinitialization process of the initial condition is outlined above for a well known linear algebraic system.

Remembering from Section 2.5.2 the basic iterative method

$$\tilde{\mathbf{y}}^{n+1} = S\tilde{\mathbf{y}}^{n} + M^{-1}\mathbf{b},\ S = M^{-1}N \tag{4.73}$$

for solving the linear algebraic system,

$$A\mathbf{y} = \mathbf{b} \tag{4.74}$$

the steady state case will be achieved when $\tilde{\mathbf{y}}^{n+1} = \tilde{\mathbf{y}}^{n}$.
It is well known that the homogeneous system

$$A\mathbf{y} = 0 \tag{4.75}$$

where A is a singular matrix, admits an eingenvector $\mathbf{v} \neq 0$.

So, applying the splitting of the matrix A (Equation (2.58)) it comes:

$$S\mathbf{v} = \mathbf{v} \tag{4.76}$$

The important assessment is that for every iteration we have an error $\mathbf{e}^n$ which we can decompose in $\mathbf{u}^n + \mathbf{v}^n$. Hence, Equation (4.73) can be written:

$$\tilde{\mathbf{y}}^{n+1} = S\tilde{\mathbf{y}}^{n} + M^{-1}\mathbf{b} + \mathbf{u}^n + \mathbf{v}^n \tag{4.77}$$

which after algebraic manipulation results

$$\tilde{\mathbf{y}}^{n+1} = S^{n+1}\tilde{\mathbf{y}}^{0} + M^{-1}\mathbf{b} + \mathbf{u}^{n} + \mathbf{v}^{n} + \sum_{r=1}^{n} S^{r}(M^{-1}\mathbf{b} + \mathbf{u}^{n-r} + \mathbf{v}^{n-r}) \tag{4.78}$$

So, when $n \to \infty$

$$\sum_{r=1}^{n} S^{r}(M^{-1}\mathbf{b} + \mathbf{u}^{n-r} + \mathbf{v}^{n-r}) + M^{-1}\mathbf{b} + \mathbf{u}^{n} + \mathbf{v}^{n} \tag{4.79}$$

will go to ∞.

This conclusion show that $\tilde{\mathbf{y}}^{n+1}$ in Equation (4.78) also goes to ∞, which is physically unacceptable.

Two solutions for this problem can be think of. One is naturally stop the computation when the successive approximations of $\mathbf{y}$, $\tilde{\mathbf{y}}^{n}$ and $\tilde{\mathbf{y}}^{n+1}$ are identical. Another possible option, which is implemented in `transient Pamg Single-phase`, is to reset the initial conditions to the values used in the beginning of the computation. This is controlled by measuring the amount of change between the solutions computed in consecutive time steps. When there is no variation in the computed solutions for consecutive time steps the steady state has been achieved. Hence, the error $\mathbf{e}^{n}$ will be cleaned before starting the new time step iterative process.

This summarises the reinitialization process of the initial condition.

Figure 4.4 shows the horizontal velocity profile of a single-phase flow through a channel, with and without the reinitialization process. It is visible that after reaching the steady state solution, the horizontal velocity after 35 time steps without the reinitialization process, drastically change to values that do not correspond to the solution of the problem.

4.4.4 Transfer Operators. Wall Treatment

The restriction and prolongation operators used by `transient Pamg Single-phase` are basically the same operators used by the steady `Pamg` single-phase solver (see Section 4.2). However, the prolongation of the velocity corrections near the walls required a special treatment. The first-order velocity corrections introduce errors into the prolongation procedure, if the patches are located near the walls. To overcome this, a Lagrange three-point[76] interpolation algorithm has been introduced to deal with the velocity corrections near the walls.

This prolongation algorithm, for the horizontal velocities can be expressed by the following formulas:

$$u^{(f)}_{if+1/2,jf} = \frac{15}{16}u^{(c)}_{ic+1/2,jc} - \frac{3}{32}u^{(c)}_{ic+1/2,jc+1} + \frac{5}{32}u^{(c)}_{ic+1/2,jc-1} \tag{4.80}$$

$$u^{(f)}_{if+1/2,jf+1} = \frac{15}{16}u^{(c)}_{ic+1/2,jc} - \frac{3}{32}u^{(c)}_{ic+1/2,jc-1} + \frac{5}{32}u^{(c)}_{ic+1/2,jc+1} \tag{4.81}$$

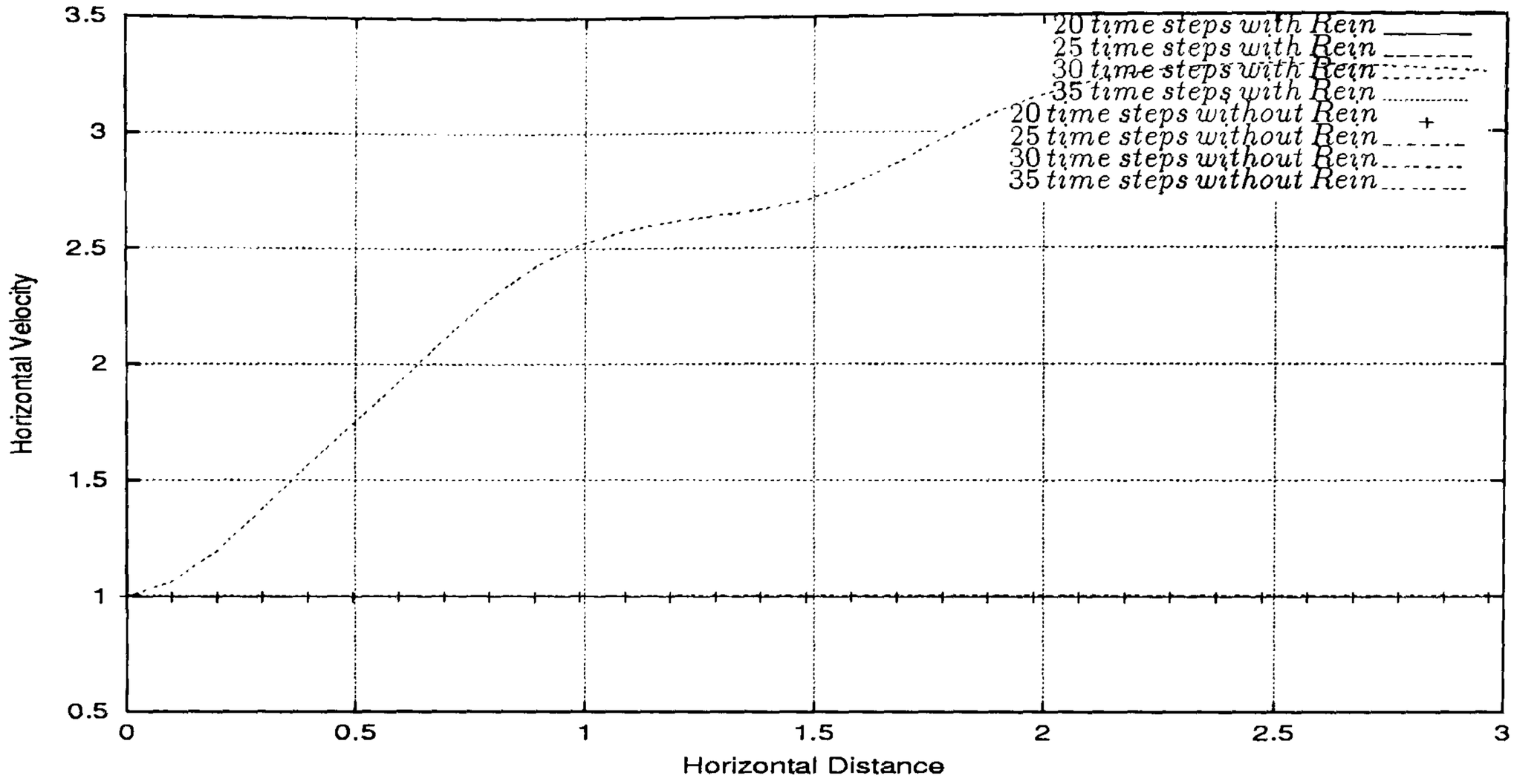


Figure 4.4: Single-phase flow through a channel - horizontal velocity profile along the line $y = 0.5$ for some time steps with and without the Reinitialization process

and for the vertical velocities by:

$$v^{(f)}_{if,jf+1/2} = \frac{15}{16} v^{(c)}_{ic,jc+1/2} - \frac{3}{32} v^{(c)}_{ic-1,jc+1/2} + \frac{5}{32} v^{(c)}_{ic+1,jc+1/2} \tag{4.82}$$

$$v^{(f)}_{if-1,jf+1/2} = \frac{15}{16} v^{(c)}_{ic,jc+1/2} - \frac{3}{32} v^{(c)}_{ic+1,jc+1/2} + \frac{5}{32} v^{(c)}_{ic-1,jc+1/2} \tag{4.83}$$

In Figure 4.5 we see the geometric interpretation of the Lagrange three-point interpolation algorithm for the horizontal velocities.

Figure 4.6 shows the geometric interpretation of the Lagrange three-point interpolation algorithm for the vertical velocities.

4.5 The transient Pamg Multi-Phase Algorithm

The focus on this section is on the `transient Pamg Multi-phase` derived from the steady `Pamg multiphase` which has been the subject of Section 4.3.

Since the steady `Pamg multiphase` has been developed from the steady `Pamg` solver there are many common features, which originates identical features in the two transient solvers that they originate.

The Symmetrical Coupled Gauss Seidel procedure is implemented in the `transient Pamg Multi-phase` like it is in the `transient Pamg Single-phase`. The Full Approximation Storage (FAS) implementation is very similar to that one implemented in the `transient Pamg Single-phase`. Moreover, the Symmetrical Coupled Gauss

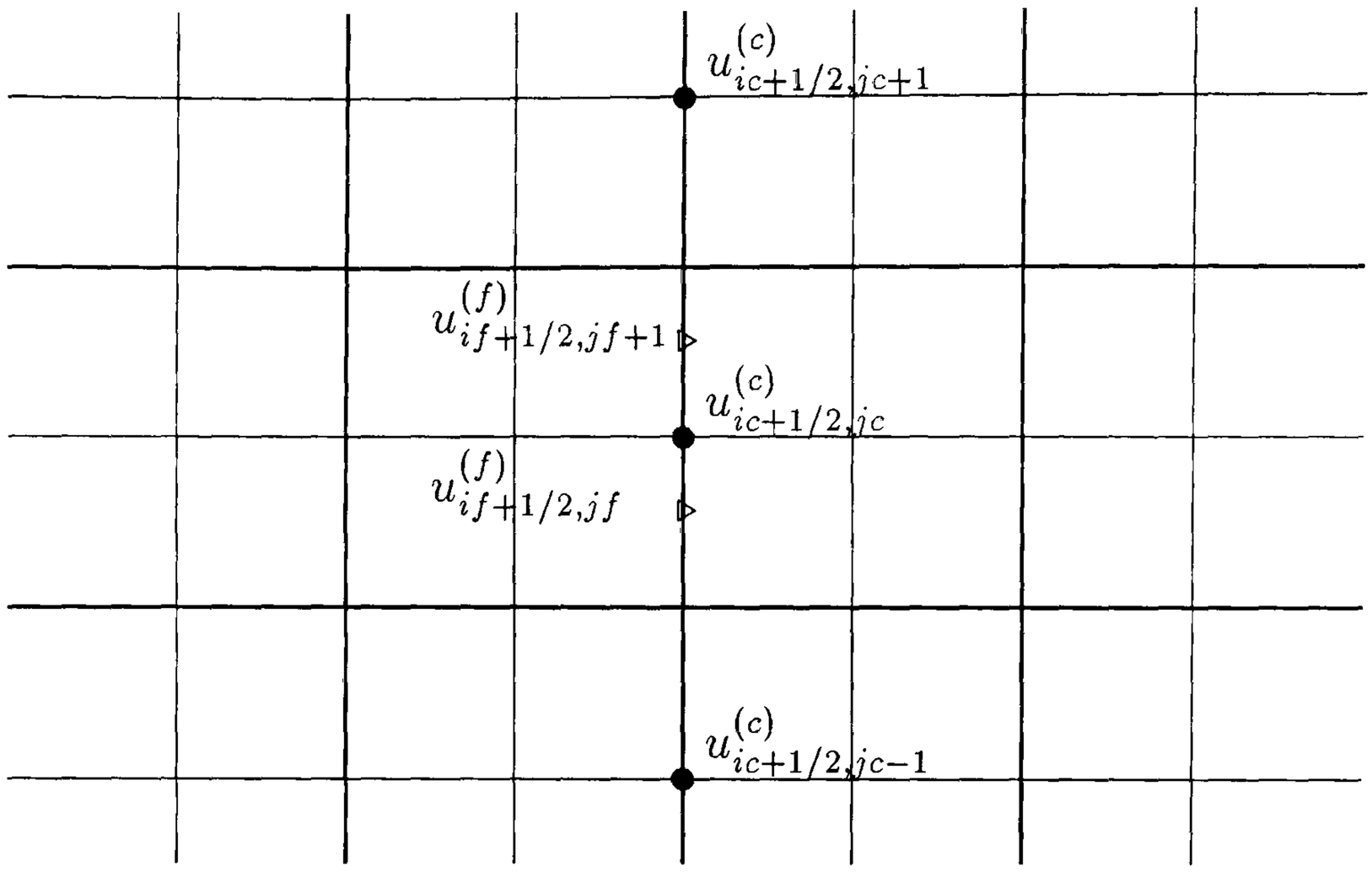


Figure 4.5: Lagrange three-point interpolation algorithm for the horizontal velocities

Seidel (SCGS) procedure is still the strategy of correcting the velocities, pressures and volume fractions. Furthermore, the presence of the volume fractions naturally originates that the system of equations to solve for each time step is much more non-linear and consequently the Newton's method is still an adequate option for the solution algorithm.

Naturally, automatic differentiation, which has been successfully validated in the `transient Pamg Single-phase`, is also implemented in this transient multi-phase solver. However, the automatic differentiation package `AD01`, was already implemented in the steady `Pamg multiphase`. So, for a question of simplicity `AD01` has been kept in the transient version. It is not in the aim of this thesis to compare the performances of `AD01` and `Adifor`.

The line searching strategy outlined in Section 4.3 has been crucial in the simulation of multi-phase flows. Strong evidence has been observed that without line searching implementation the transient solver does not achieve convergence, when simulating a two-phase fluid flow through a channel.

The inter-grid transfer operators, namely the restriction and the prolongation operators are the same as those used by the steady `Pamg multiphase` solver. No additional treatment has been identified for any particular subdomain of all the test cases studied.

4.6 Conclusions

In this Chapter, the algorithms implemented in `transient Pamg Single-phase` and `transient Pamg Multi-phase` have been described. Since they were born

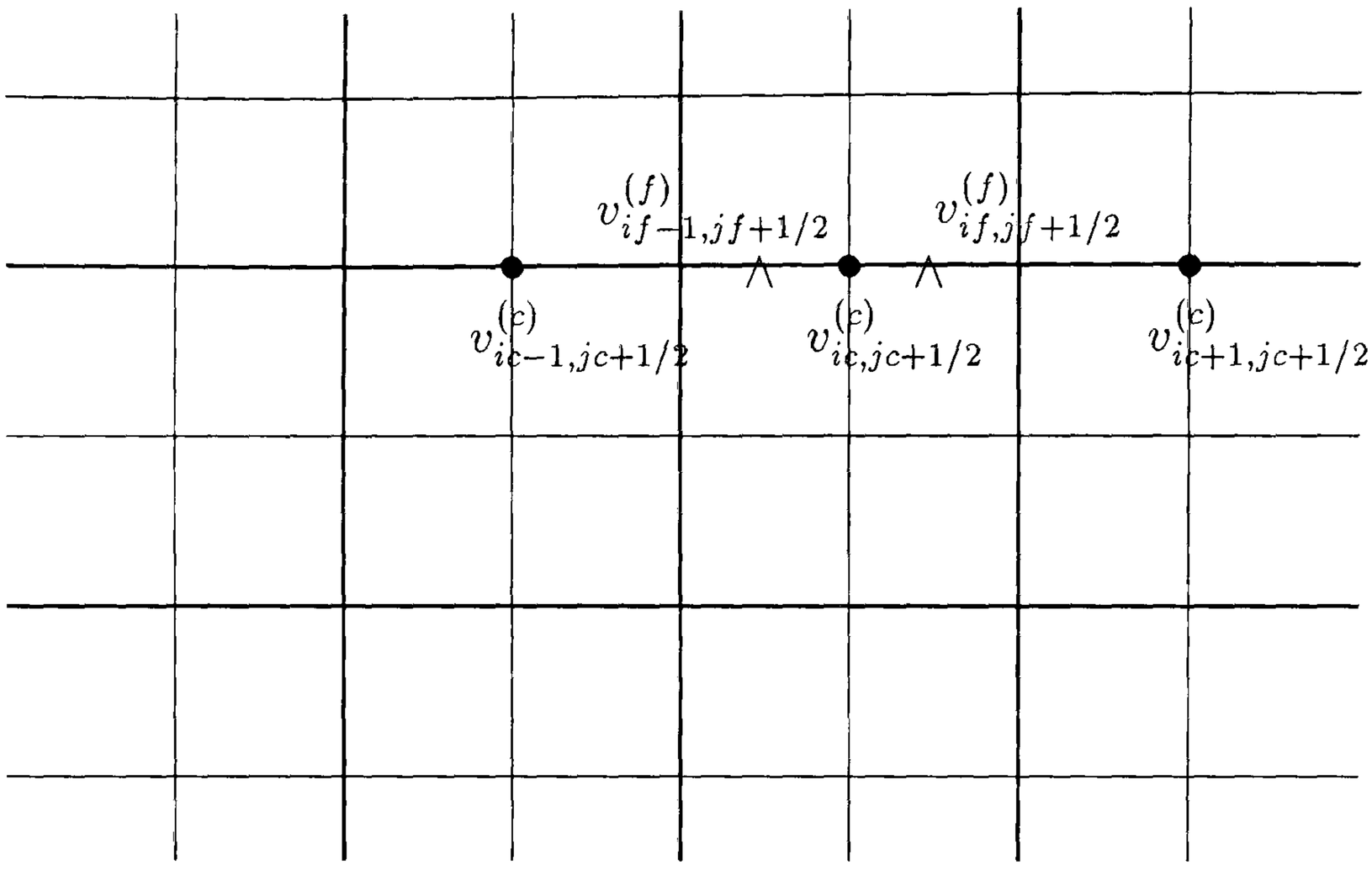


Figure 4.6: Lagrange three-point interpolation algorithm for the vertical velocities

from the equivalent steady solvers, some focus on the main features of the steady `Pamg` and `Pamg multiphase` has been done.

All the four solvers are very similar but the presence of the time dependent terms in the transient codes originate some new features to deal with. Even in the single-phase transient code, it has been necessary to implement Newton's method in order to get convergence to the solution with a reasonable amount of work. Furthermore, the use of an automatic differentiation package has been necessary to obtain the expressions for the Jacobians. Another important issue, which has been discussed, is the initial guess implementation process. The necessity of a reinitialization process of the initial condition has been demonstrated. This was due to the necessity of bounding the error of each iteration in order to achieve a stable iterative scheme after reaching the steady state solution.

Naturally, there was a cost for some of the developments outlined. Since for the transient solvers we have to solve the system of unknowns after each time step, the use of automatic differentiation packages degrades considerably the run time of the solvers. Evidence suggests that automatic differentiation can account for up to 70% of the computation time. One possible optimisation could be the replacement of the automatic differentiation packages by routines specially implemented for the computation of the local Jacobians. This was not in the scope of this thesis. The implicit method implemented in both `transient Pamg Single-phase` and `transient Pamg Multi-phase` let us increase the time step with a reasonable value. So, the run time of the code at the final time of integration can diminish considerably.

Chapter 5

Transient PAMG. Validation on Uniform Grids

5.1 Introduction

The aim of this chapter is to establish the correctness of the implementation of `transient Pamg Single-phase` and `transient Pamg Multi-phase` solvers. This is established by comparing the results given by the solvers with the commercial CFD code, `CFX 4.3`[21], which has been extensively validated and is widely used. Furthermore, comparisons with the analytic solutions, when available are also outlined.

The accuracy of the `transient Pamg Single-phase` solver is established using three validation cases:

- Firstly, a single-phase channel flow
- Secondly, a single-phase flow through an expansion
- Thirdly, a single-phase flow through a contraction

For the validation of the `transient Pamg Multi-phase`, three multi-phase cases of varying complexity, have been chosen:

- a two-phase flow through a channel
- a two-phase flow through a T-junction with two inlets
- a two-phase flow through a contraction.

The presentation of the results is done choosing particularly difficult regions of the domain. Focus on the degree of grid-independence and time-step independence is discussed. The grid-independence suggests that the solution is not significantly dependent on the size of the discretization grid, and can be considered to be an accurate representation of the continuous solution. However the volume fractions fields are quite sensitive to the grid size. Since the solutions given by `CFX 4.3` also display this behaviour this must be due to the characteristic of the governing equations.

After having established the accuracy of `transient Pamg Single-phase` and `transient Pamg Multi-phase` the emphasis has been on the performance of the two solvers. The robustness (see Chapter 1 for the definition) of the solution algorithms is established by verifying that convergence is not dependent on the quality of the initial guess. The choice of the initial guesses for starting the simulations has been as general as possible. Initial pressure and velocities distributions were set to zero. The volume fractions set to 0.5 for each phase of the multi-phase regime.

The numerical convergence of the solvers is also studied. The convergence rates are not entirely grid independent. This phenomenon is observed in the simulation of multi-phase fluid flows. The high complexity and strong non-linearity of the governing equations cause this. Furthermore, solutions on different grids can be quite different from each other. The cost, measured in terms of work units, is also calculated for the simulation of fluid flows in the several test cases with different number of grid levels.

5.2 Transient Pamg Single-Phase. Validation Problems

5.2.1 Single-Phase Channel Flow

Problem definition

In this first test case, a flow of a single-phase fluid in a channel is considered. The physical properties chosen are:

Flow Regime	Reynolds Number	Viscosity	Density
Single-Phase	1000	0.01	1.0

The Reynolds Number is defined by[77]:

$$Re = \frac{uh}{\nu} \tag{5.1}$$

where u is the horizontal velocity in the inlet, h is the height of the domain in the inlet and ν the kinematic viscosity.

Figure 5.1 shows the computational domain chosen for this simulation. It is just a simple rectangle with $0 \leq x/h \leq 3$ and $0 \leq y/h \leq 1$. The resolution of the coarsest grid is $\Delta_{xsize} = \Delta_{ysize} = 0.25$. In Figure 5.2 we see a grid at the uniform level 3. Figure 5.3 shows the sections where the solution profiles are going to be presented.

The boundary conditions used are at the inlet, Dirichlet boundary conditions and at the outlet Neumann conditions. In the inlet a parabolic velocity profile is

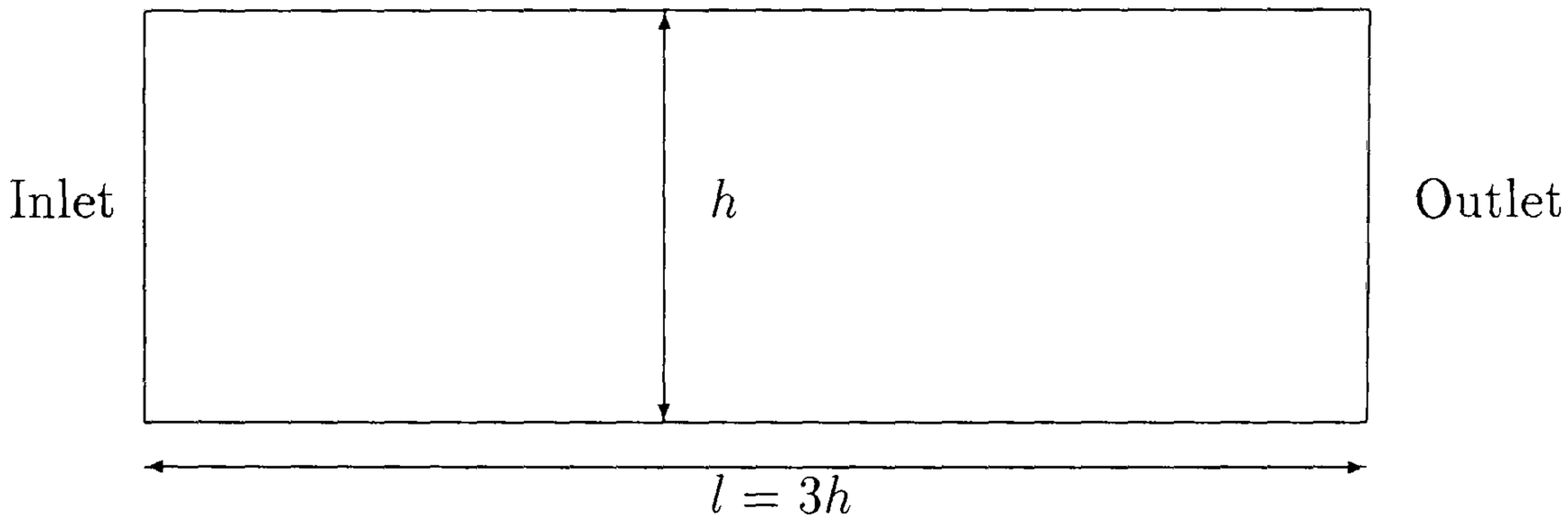


Figure 5.1: Geometrical representation of the domain of the Channel flow problem

specified for the x-component velocity $u(x, y)$ by

$$u(0, y) = 4y(1 - y)$$

and for the y-component velocity $v(x, y)$ it is null in the inlet boundary ($v(0, y) = 0$). At the outlet the y-component velocity is also null.

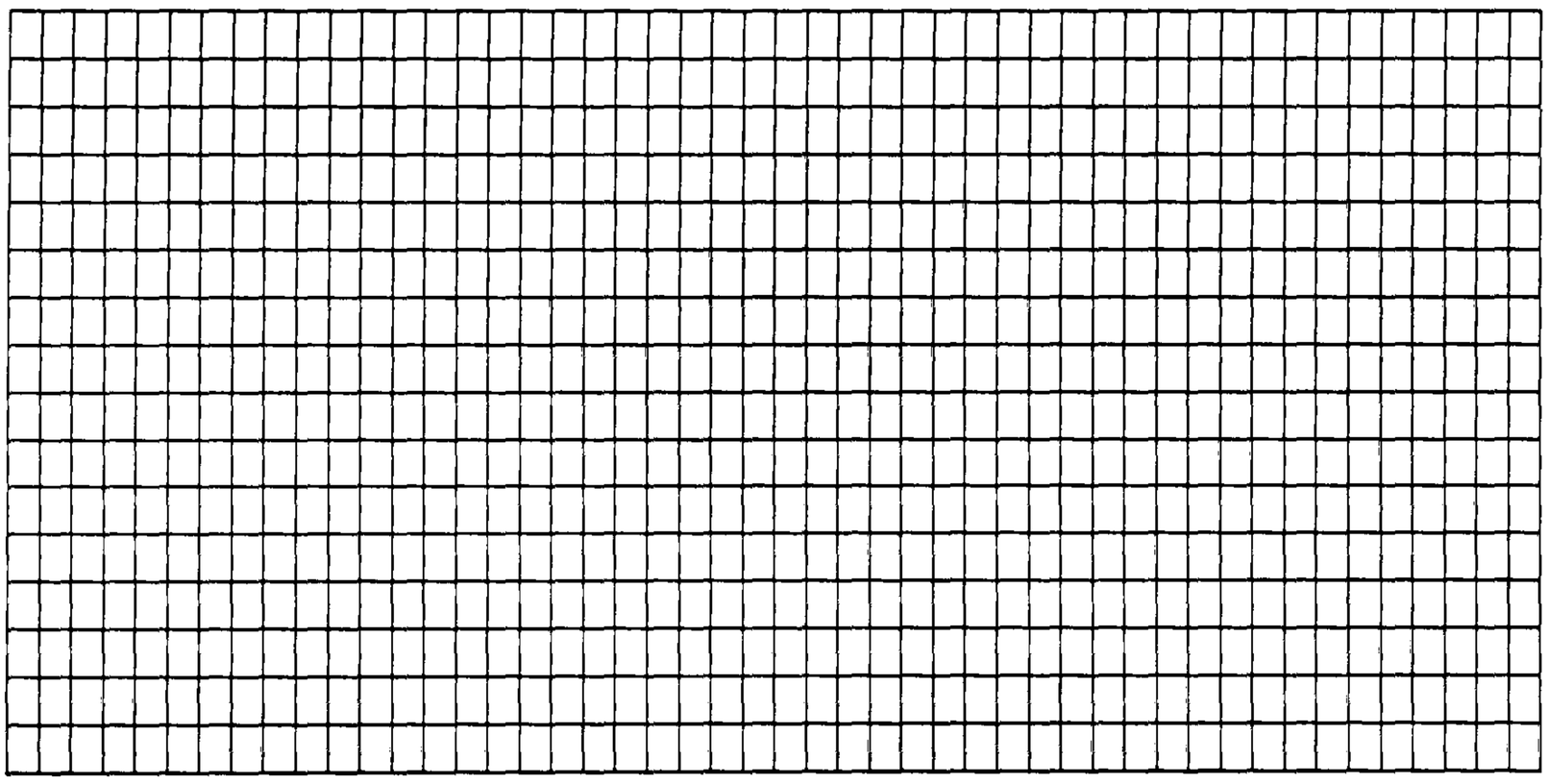


Figure 5.2: Uniform level 3 grid for the Channel flow test case ($\Delta_{xsize} = \Delta_{ysize} = 0.0625$)

The initial condition for the simulation is very simple. The velocities are all set to zero. This ensures that the solver does not depend on an initial guess very close to the solution and consequently establishes the large domain of convergence of the code. The convergence tolerance for the simulation is 10^{-6} for each time step. Since the solver is a non-dimensional one we use the concept of time unit to specify the simulation at a specific time of integration.

Results for a 3 level computation

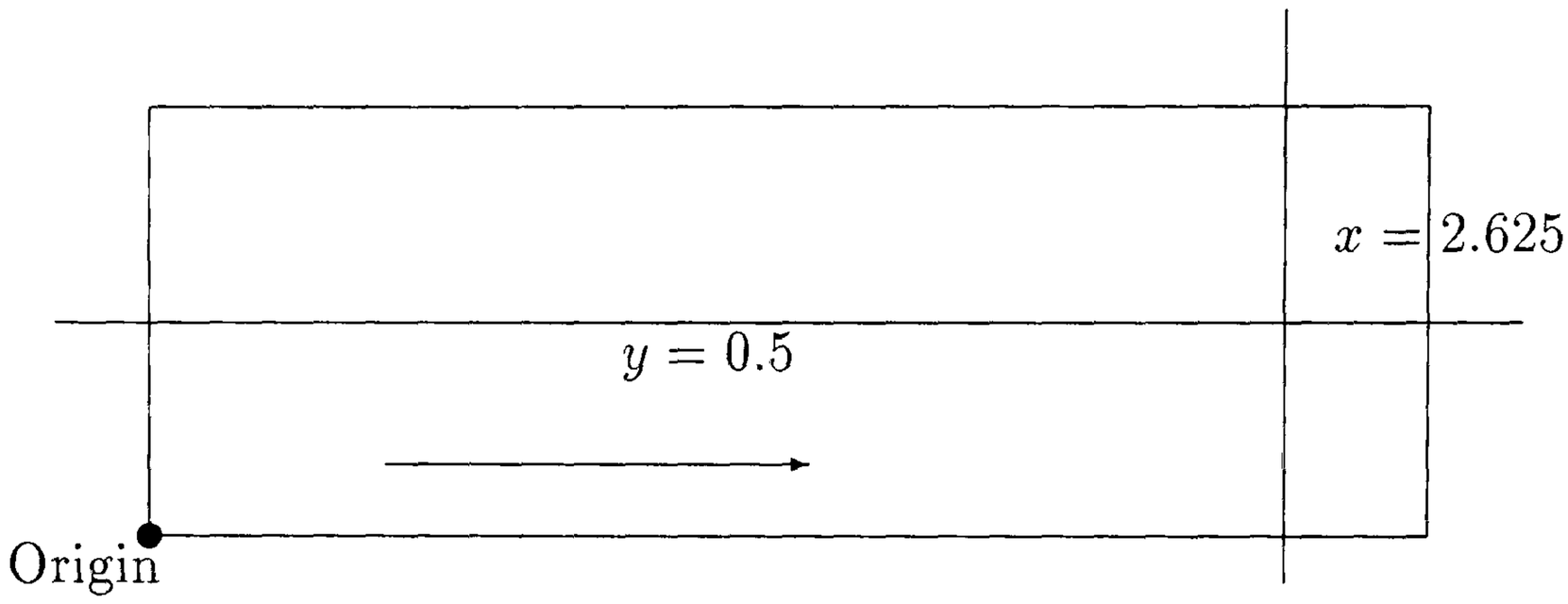


Figure 5.3: Channel Flow test case - main sections used to give the solution profiles

In Figure 5.4 we see the streamlines computed for this test case, after 20 time units. Since the vertical velocities are identically zero everywhere for this geometry, the streamlines are straight horizontal lines. The aspect ratio that can be seen on this picture has been distorted only for graphical reasons.

The parabolic velocity profile along the pipe is the final solution after some time of integration (see Figure 5.5). It is visible that as the time increases the fully developed profile appears. The not null values in the walls are due to interpolation errors of the plotting routine. The vertical velocity profile, presented in Figure 5.6 tends to zero after the same time of integration.

In Figure 5.7 we see the pressure drop profile along the line $y = 0.5$, after 20 time units. As we could expect it is just a straight line starting from the biggest value in the inlet and ending in the smallest one in the outlet.

Grid and Time Step Independence of the Solutions

Studies have been carried out that support the conclusion that the results provided by the solver have a significant degree of grid and time step independence. In Figures 5.8 and 5.9 we see the results for single 2 and 3 level computations over the domain after 20 time units. In Figure 5.9 the small differences that we see in the vertical velocity profiles must be due to the interpolation errors between the grids. However, the overall values of the vertical velocities are very close to zero, in all the grid cases.

In Figure 5.10 we present for two different time steps, 0.1 and 1.0, the velocity profiles after 1, 10 and 20 time units. Good agreement can be seen in the velocity profiles at 10 and 20 time units. Moreover the expected developed parabolic profile is visible at this time. This suggests that the solver does not depend on the time step size.

Comparison with `CFX 4.3` Solutions

Figures 5.11 through 5.13 present the comparison results between `transient`

`Pamg Single-phase` and the commercial code `CFX 4.3`. The pressure drop profiles after shifting demonstrate good general agreement (see Figure 5.11). However, in the inlet and outlet some differences are visible. This may be due to different discretization of the governing equations and different gridding used by `CFX 4.3`. The horizontal velocity profiles (see Figures 5.12 and 5.13) exhibit the same pattern. The small differences in the horizontal velocity along the line $x = 2.625$ are due to the different number of cells in the computational domains for `transient Pamg Single-phase` and `CFX 4.3`.

Single-Phase Channel Flow with an accelerating horizontal velocity in the inlet

A variation of the single-phase channel flow with an accelerating horizontal velocity in the inlet, has been also simulated. The single-phase fluid flow has exactly the same physical properties. The computational domain is exactly the same shown in Figure 5.1. However, in the inlet, a parabolic velocity profile is specified for the x-component velocity $u(x, y)$ by

$$u(0, y) = ky(1 - y)$$

where k is a parameter given by

$$\begin{cases} k = time * 0.4 + 0.4 & \text{for } time \leq 9 \\ k = 4.0 & \text{otherwise} \end{cases}$$

and for the y-component velocity $v(x, y)$ it is null in the inlet boundary $(v(0, y) = 0)$ and at the outlet. The convergence tolerance for the simulation is 10^{-6} for each time step.

Figure 5.14 shows the horizontal velocity profile at the inlet for this test case. The profiles starts with a curve where the maximum horizontal velocity is 0.2 and ends with the curves corresponding to the range 10 to 20 time units where the maximum horizontal velocity is 1.0. In Figure 5.15 the corresponding horizontal velocity profiles at the outlet are presented. We can observe that after 20 time units the fully developed parabolic profile is similar to the inlet parabolic profile at 20 time units. This shows that the accelerating characteristics of the flow do propagate through the entire channel. Figure 5.16 shows the horizontal velocity profile along the line $y = 0.5$. It is visible the inlet acceleration from 0.2 to 1.0 and the horizontal velocity parallel curves which indicate that the acceleration is propagated over the entire channel in a very effective and equivalent manner.

Summary

With this test case we started the validation of the `transient Pamg Single-phase` solver. The choice of this channel geometry was due to its simplicity. Hence it was an ideal first test case to verify the first results produced by the solver. These exhibit the expected patterns and consequently we can conclude that the results are correct. Furthermore some tests were undertaken to verify the grid independence of the multigrid algorithm. The results have been presented and good agreement can be verified for single 2 and 3 level calculations. Very similar results have been

obtained with two different time steps. Hence the grid and time step independence of the solver could be verified.

Another experiment with an accelerating fluid flow in the inlet has been outlined. The accelerating velocities propagate through the channel in a very equivalent manner.

Finally to complement the validation process the results produced by the `transient Pamg Single-phase` have been compared with the results computed by the commercial code `CFX 4.3`. Good agreement between the two solvers could be observed.

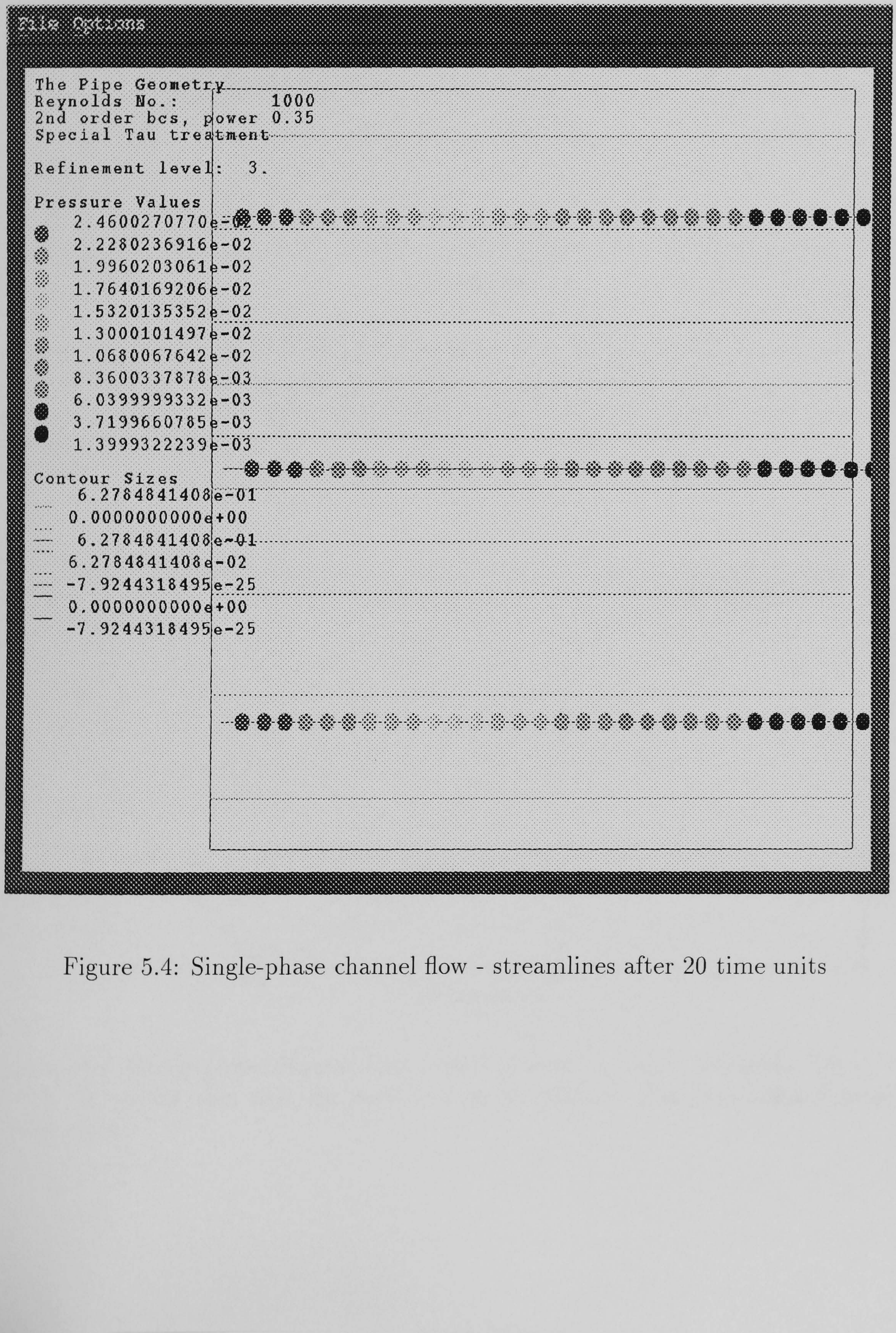


Figure 5.4: Single-phase channel flow - streamlines after 20 time units

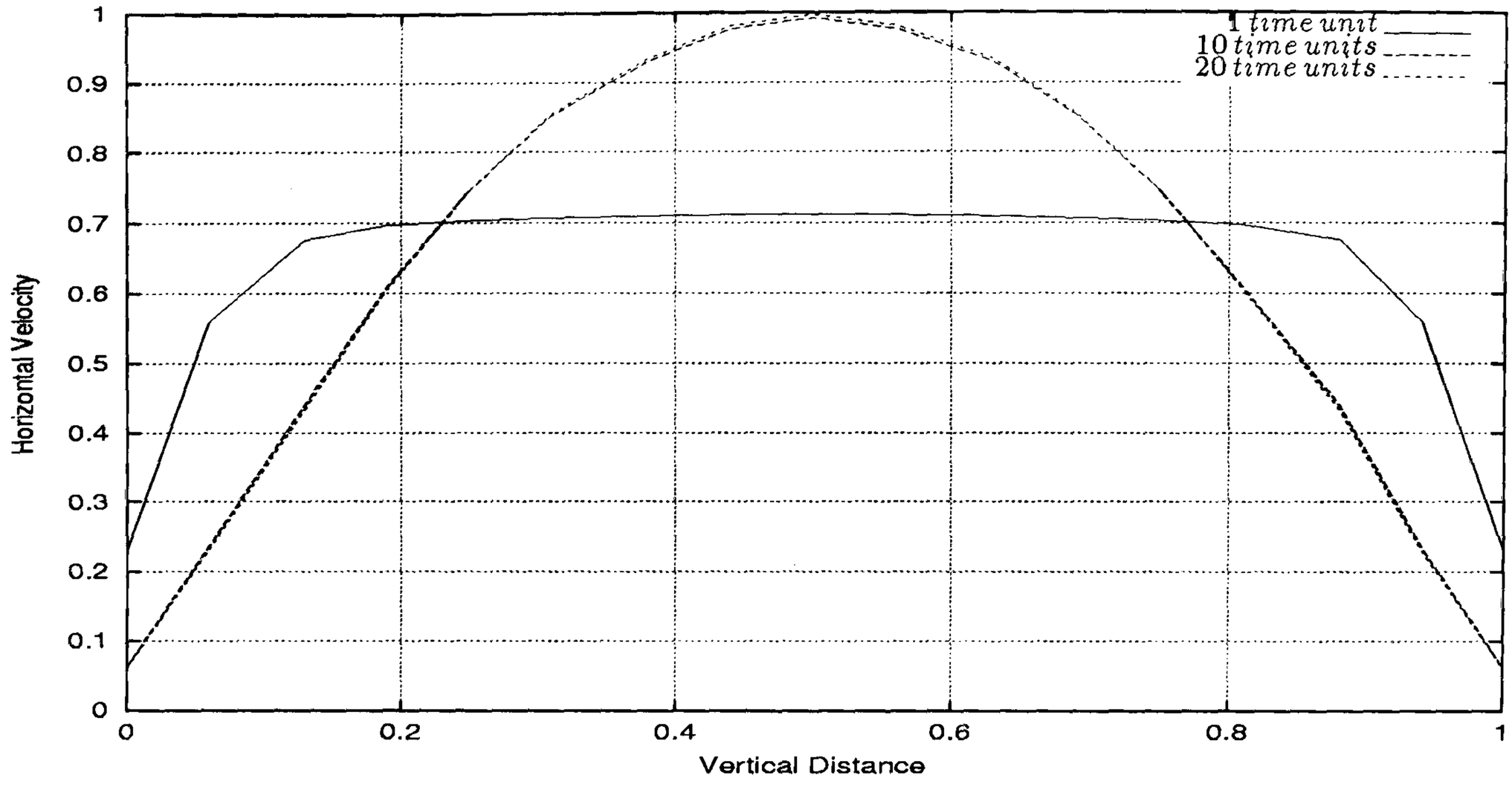


Figure 5.5: Single-phase channel flow - horizontal velocity profile along the line $x = 2.625$. It can be seen that as the time increases the fully developed parabolic profile appears. The positive values in the walls are due to interpolation errors of the plotting routine.

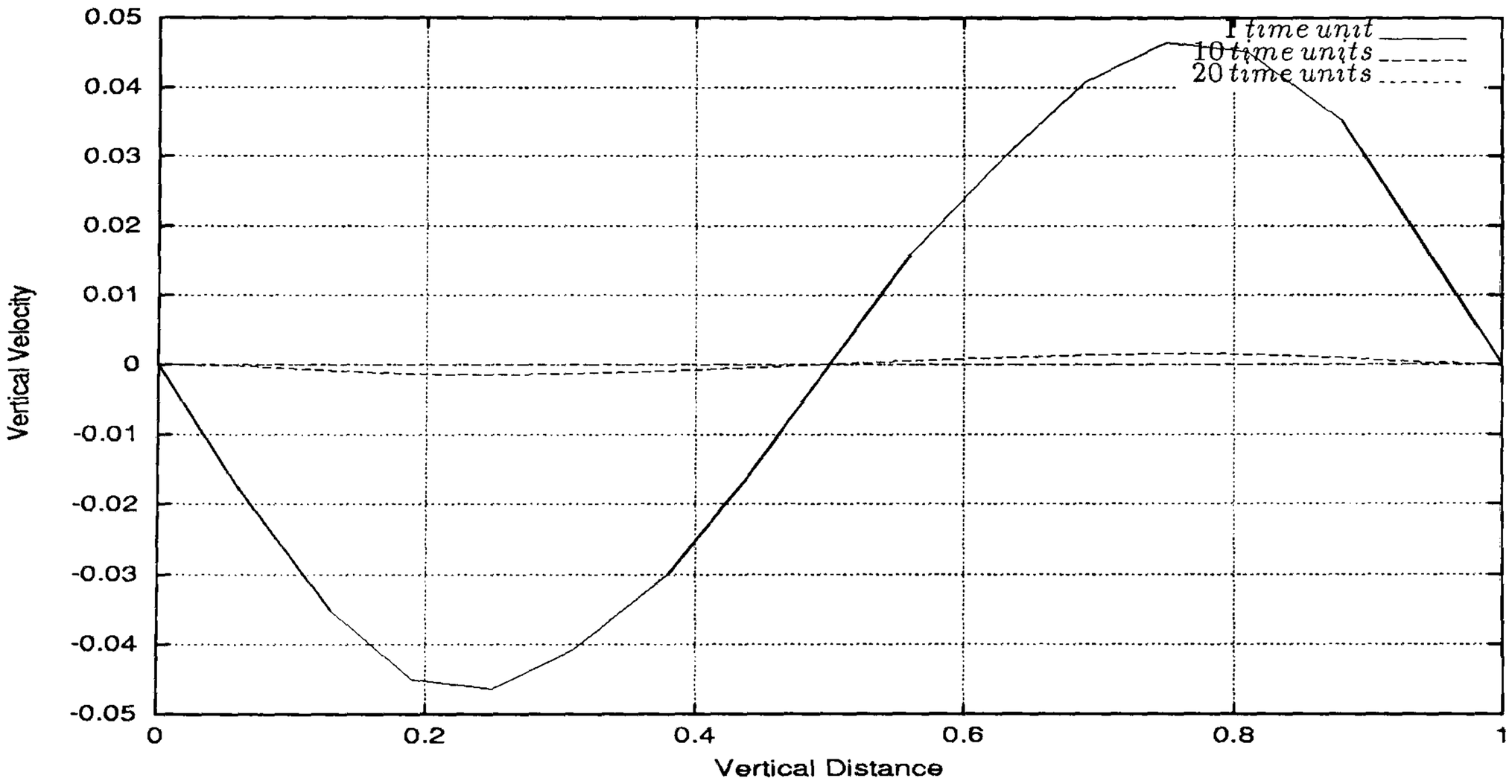


Figure 5.6: Single-phase channel flow - vertical velocity profile along the line $x = 2.625$. It can be seen that the vertical velocity tends to zero after some time of integration.

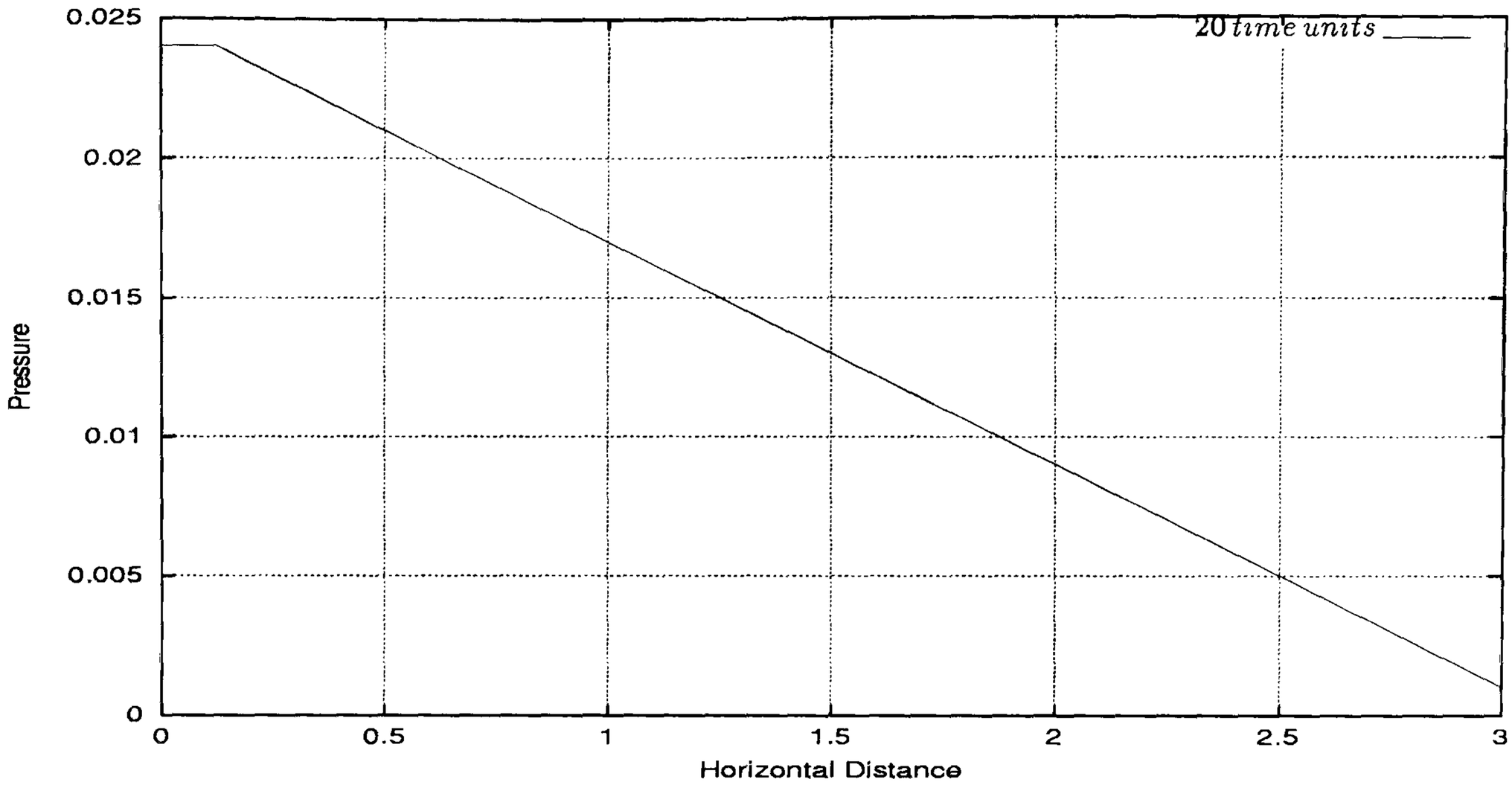


Figure 5.7: Single-phase channel flow - pressure drop along the line $y = 0.5$ at 20 time units. It is just a straight line starting from the biggest value in the inlet and ending in the smallest value in the outlet.

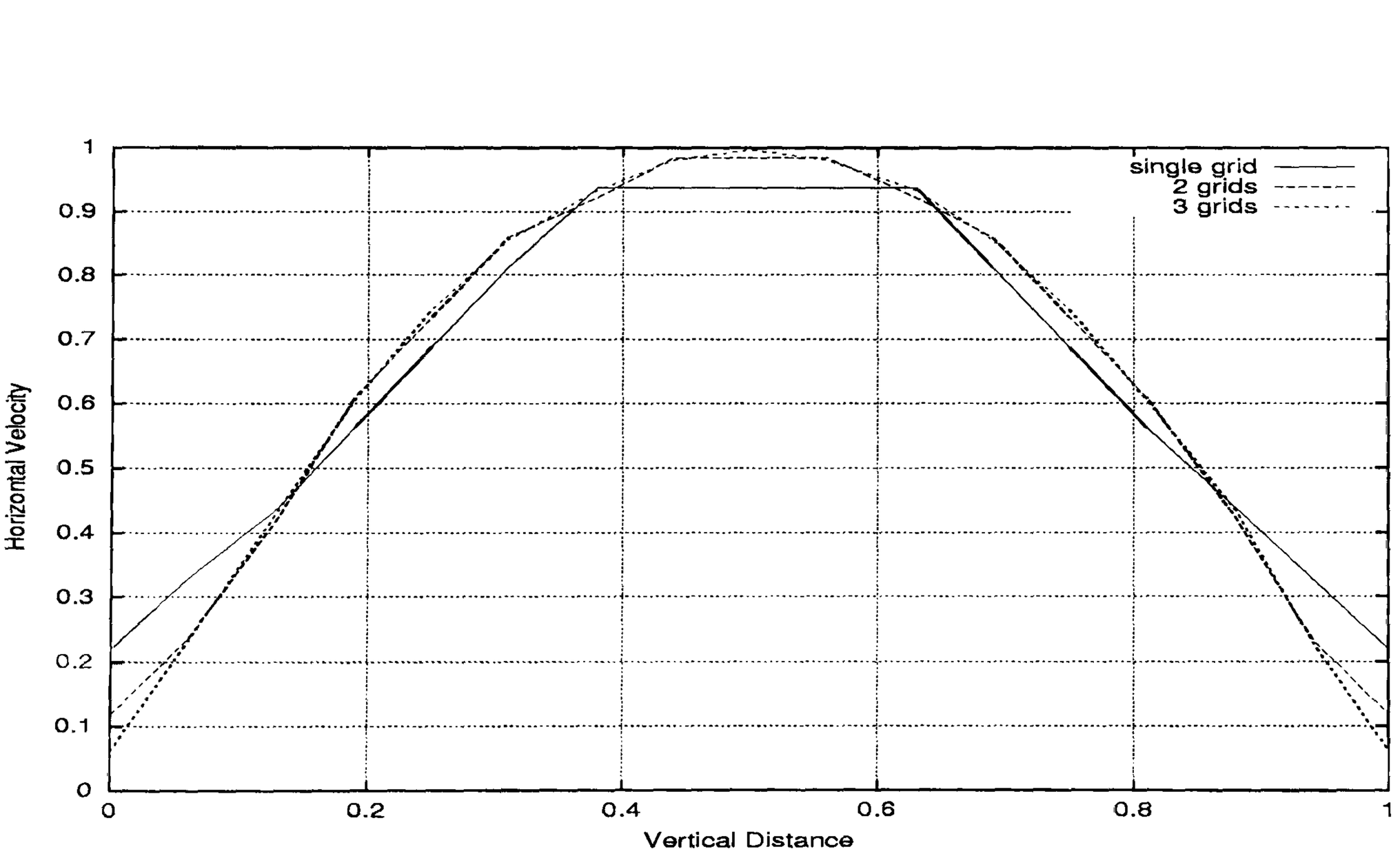


Figure 5.8: Single-phase channel flow - horizontal velocity profile for different grids along the line $x = 2.625$ after 20 time units. It can be seen that for 2 and 3 level computations the horizontal parabolic profile is fully developed.

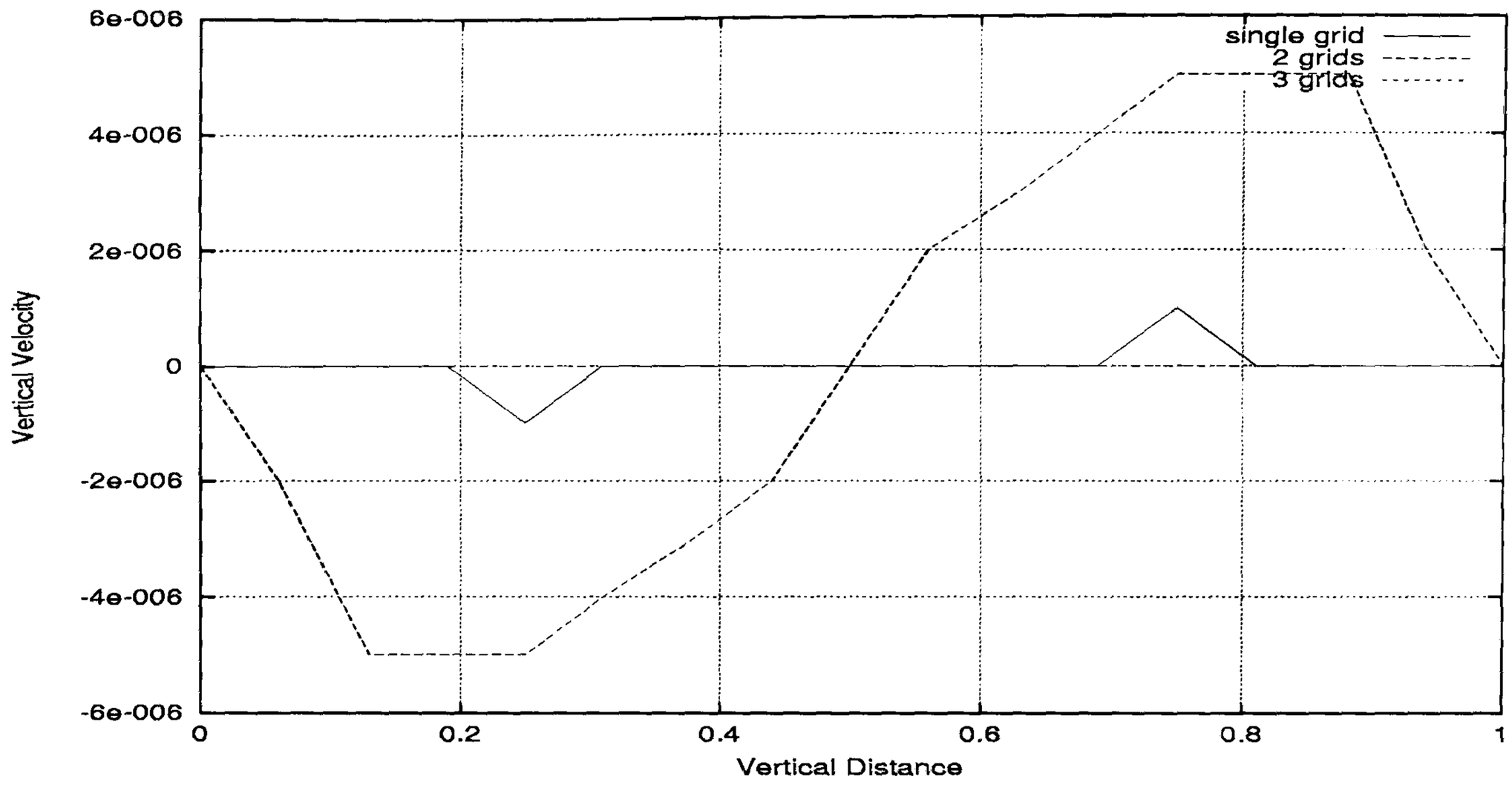


Figure 5.9: Single-phase channel flow - vertical velocity profile for different grids along the line $x = 2.625$ after 20 time units. It can be seen that the vertical velocities are very close to zero in all grid cases.

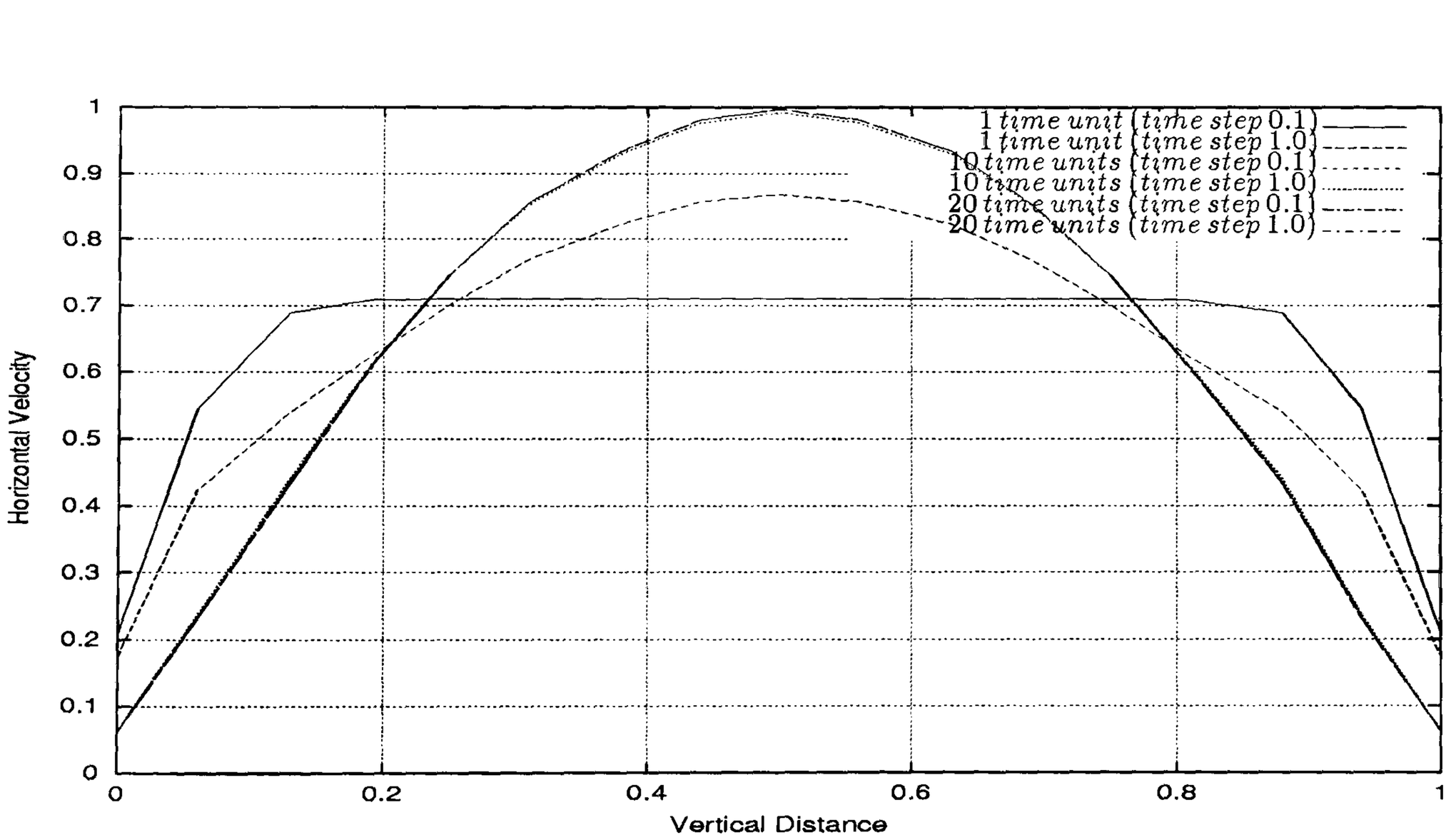


Figure 5.10: Single-phase channel flow - horizontal velocity profile along the line $x = 2.625$ for different time steps. It can be seen good agreement in the velocity profiles at 10 and 20 time units. The fully developed parabolic profile is visible at these times of integration.

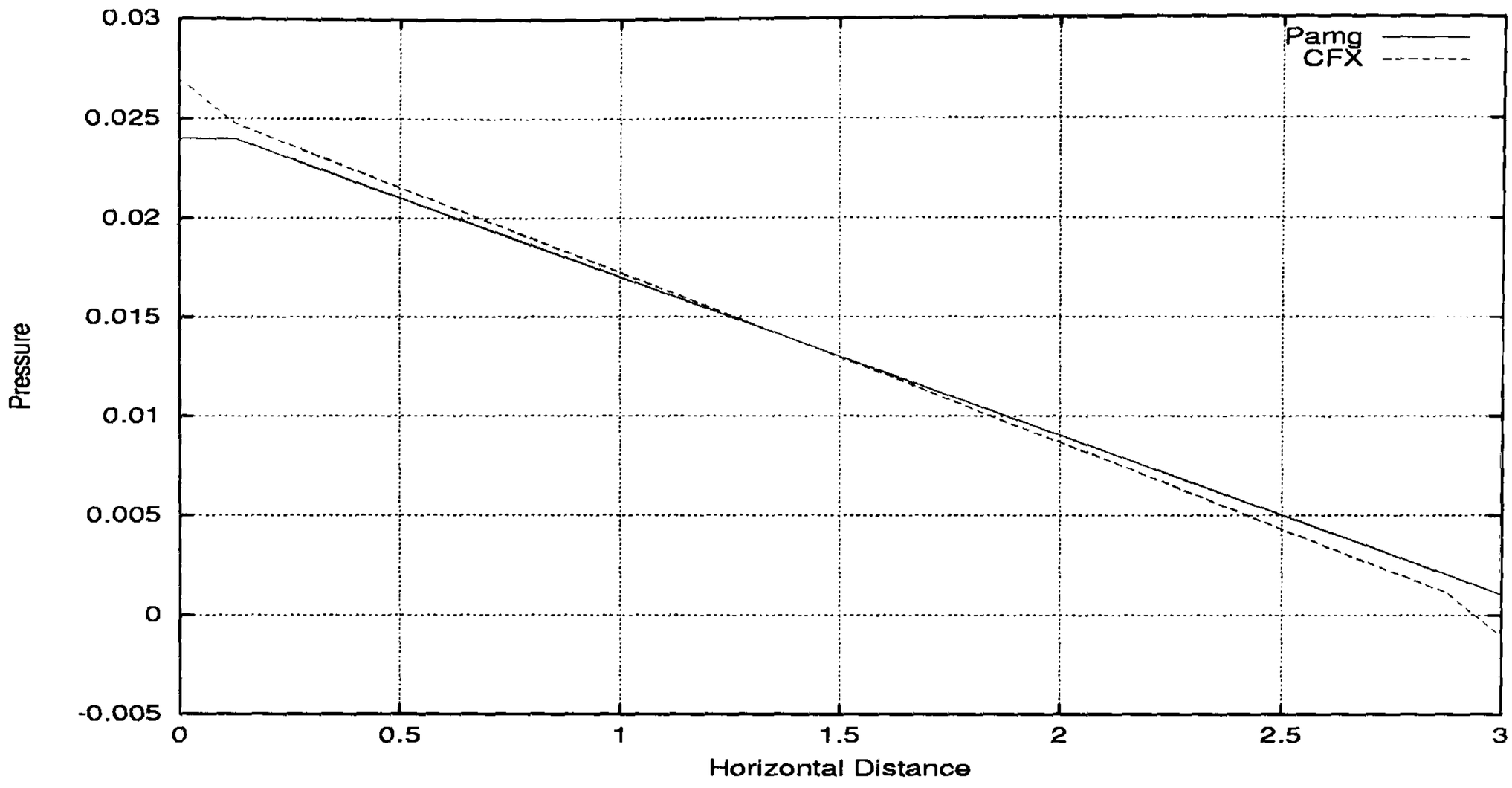


Figure 5.11: Single-phase channel flow - pressure drop along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be seen good agreement between the curves produced by both solvers.

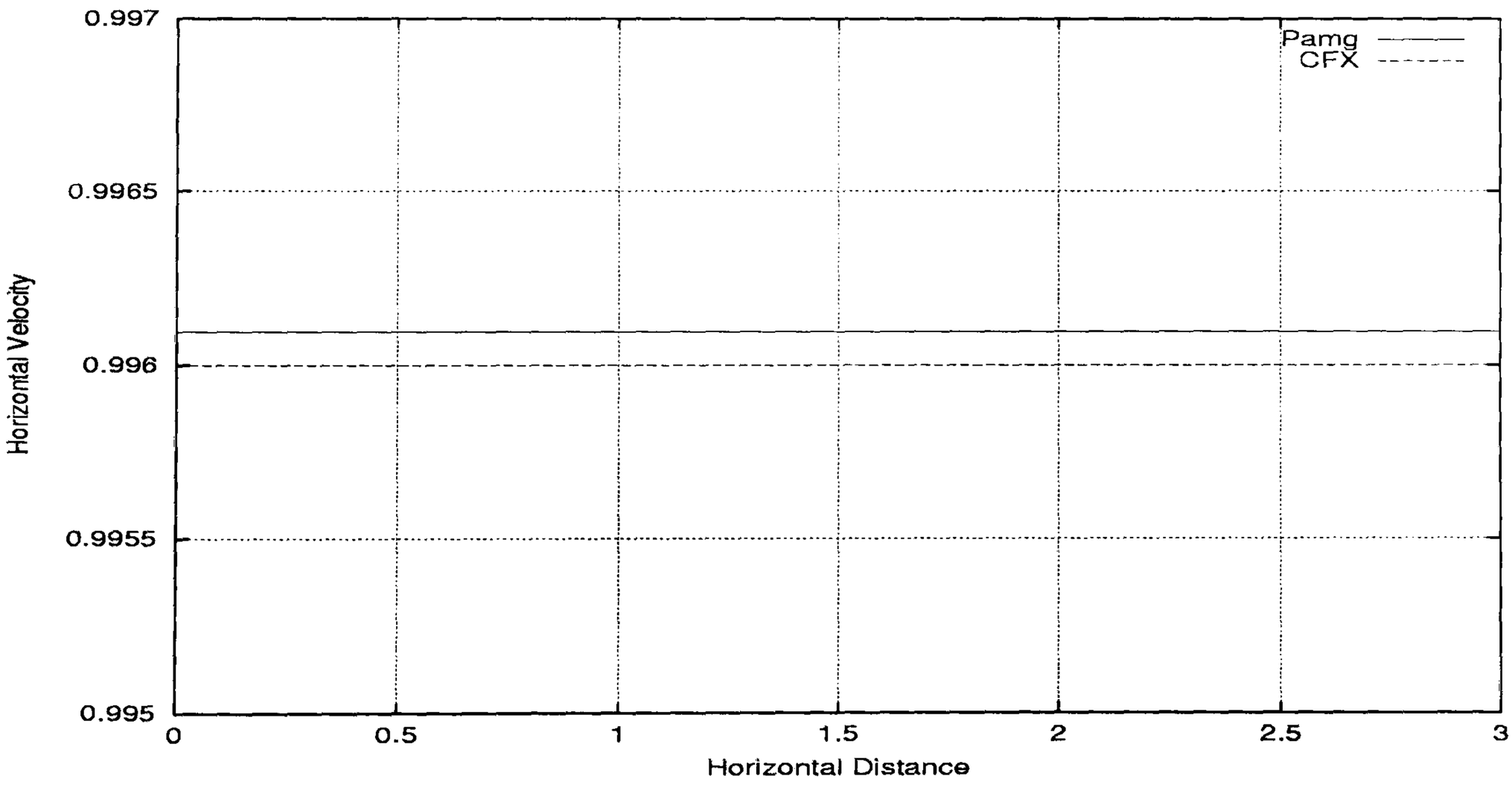


Figure 5.12: Single-phase channel flow - horizontal velocity along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be observed that the two straight lines are parallel and very close. Note the small scale of the horizontal velocity axis.

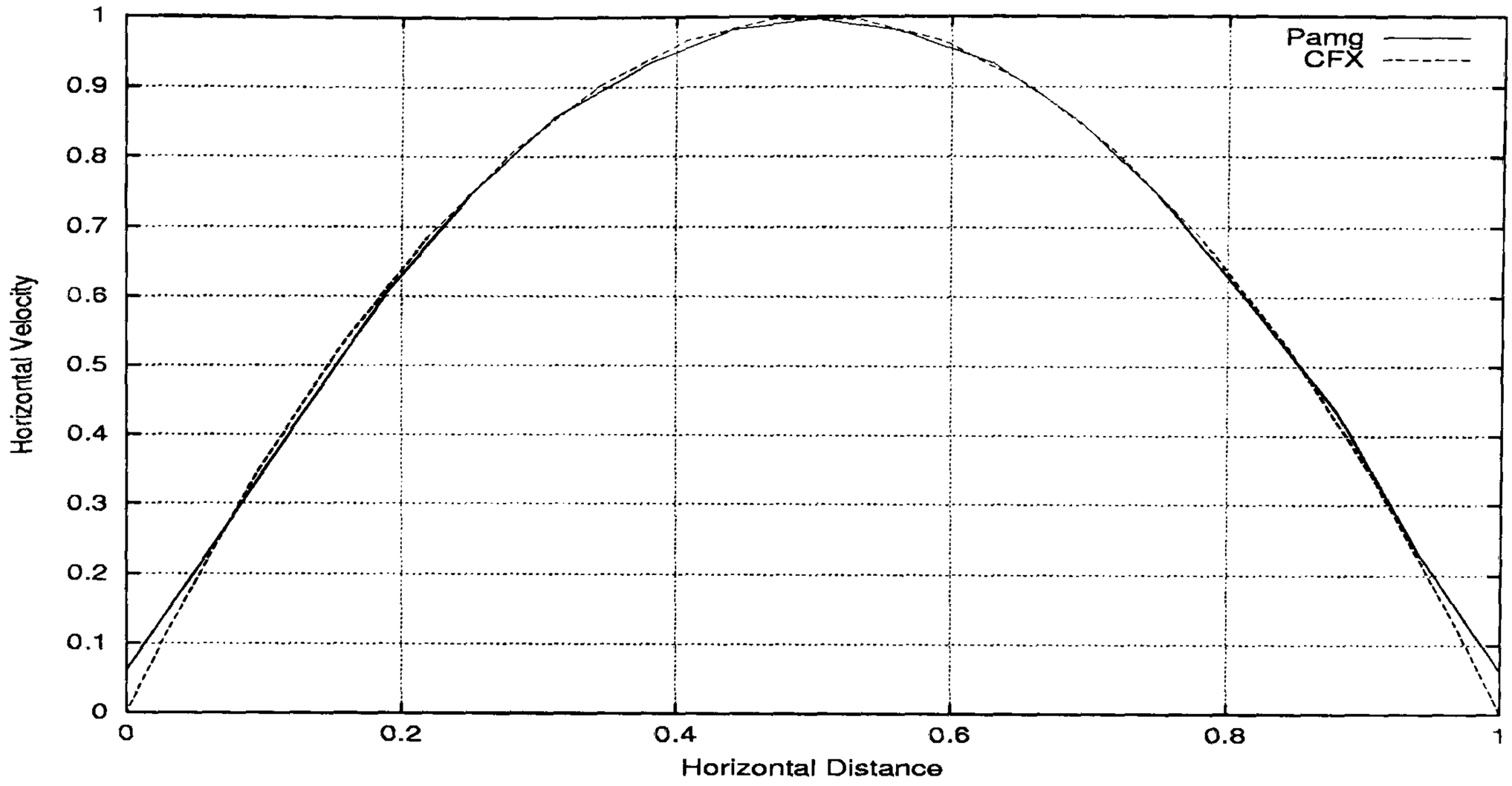


Figure 5.13: Single-phase channel flow - horizontal velocity along the line $x = 2.625$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be seen good agreement between the fully parabolic developed profiles produced by both solvers.

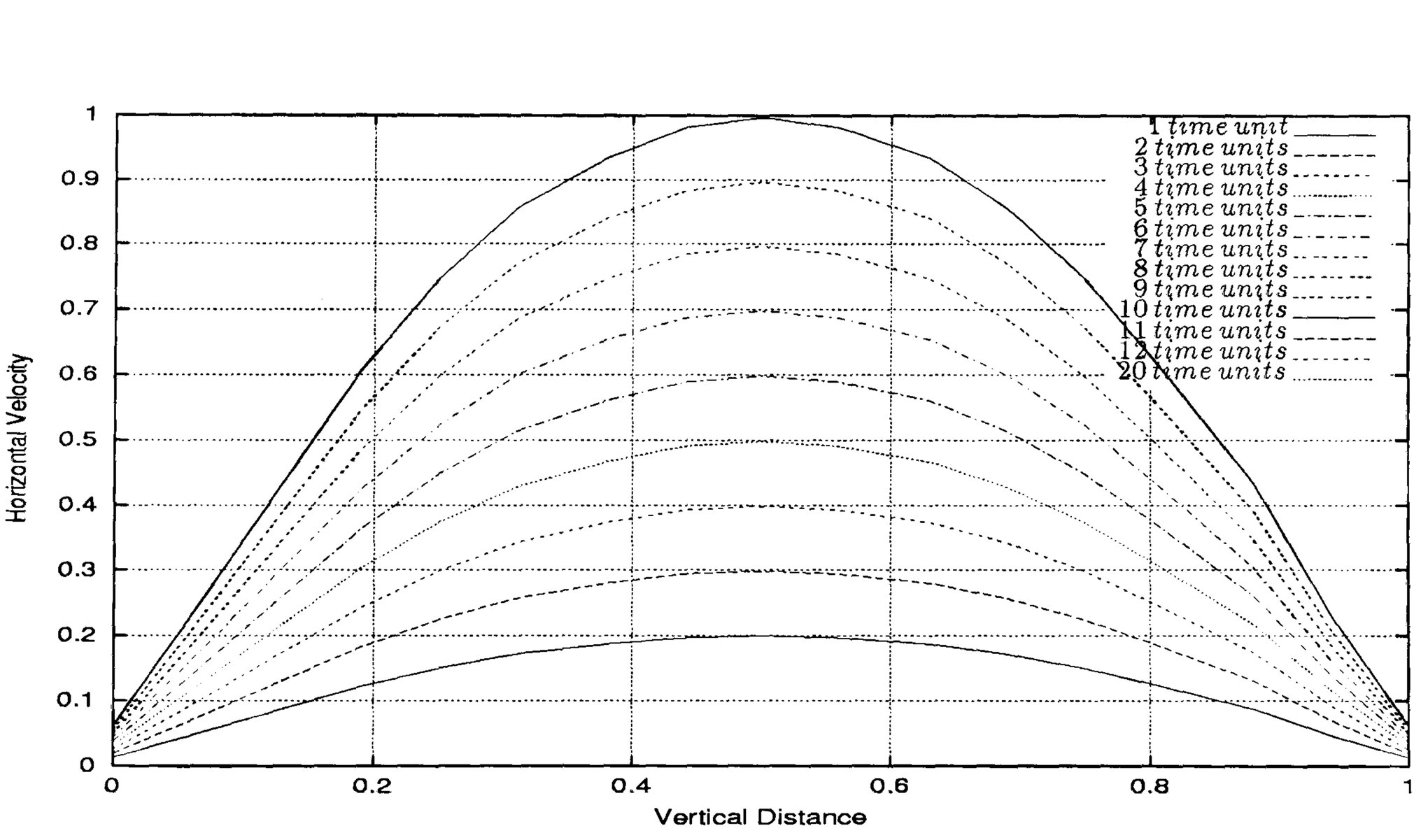


Figure 5.14: Single-phase channel flow - horizontal velocity profile at the Inlet for an accelerating fluid flow. It can be seen the growing parabolic curves starting from the maximum horizontal velocity 0.2 and ending with the maximum horizontal velocity 1.0.

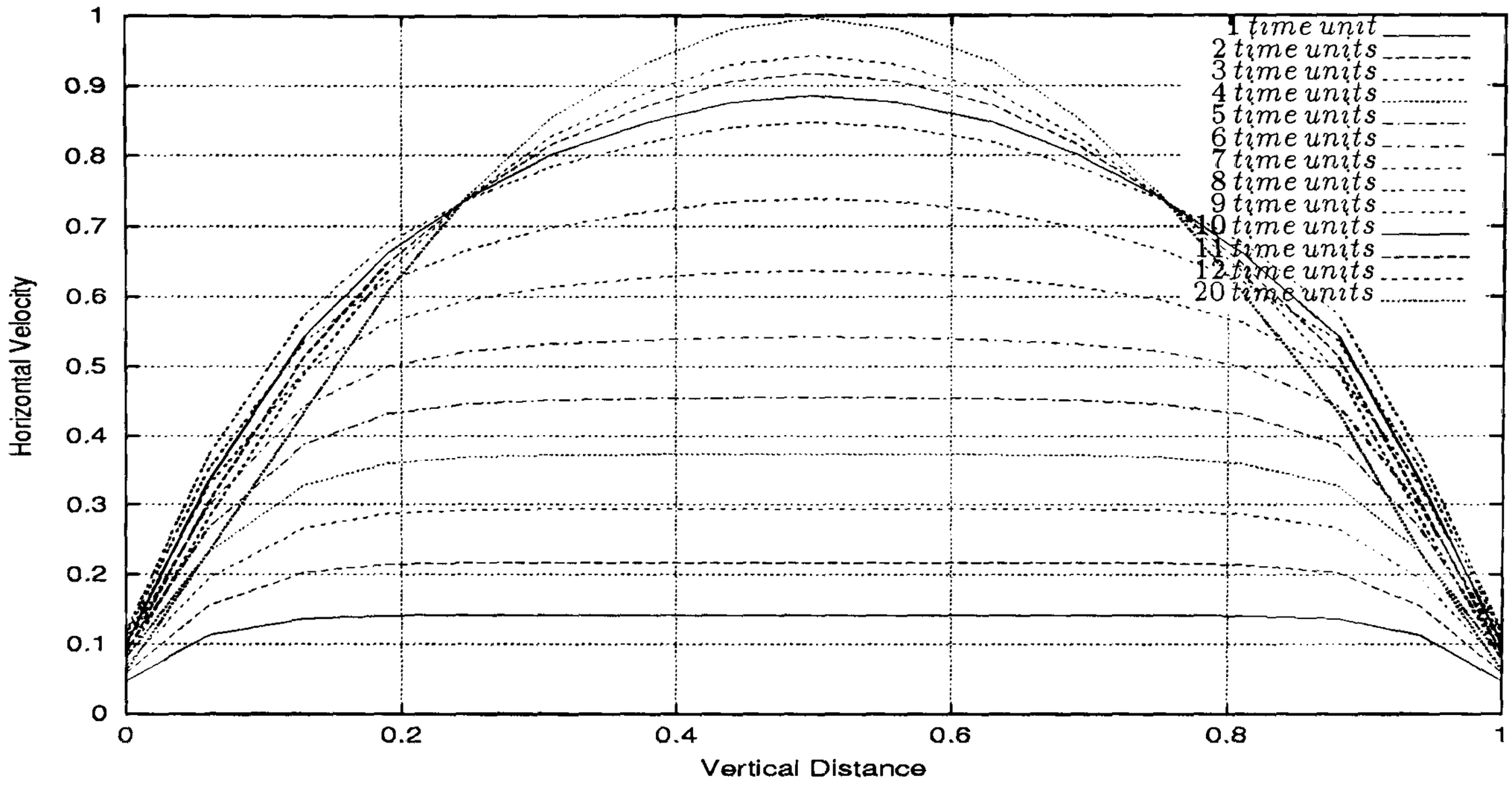


Figure 5.15: Single-phase channel flow - horizontal velocity profile at the Outlet for an accelerating fluid flow. It can be observed the fully parabolic developed profile at 20 time units.

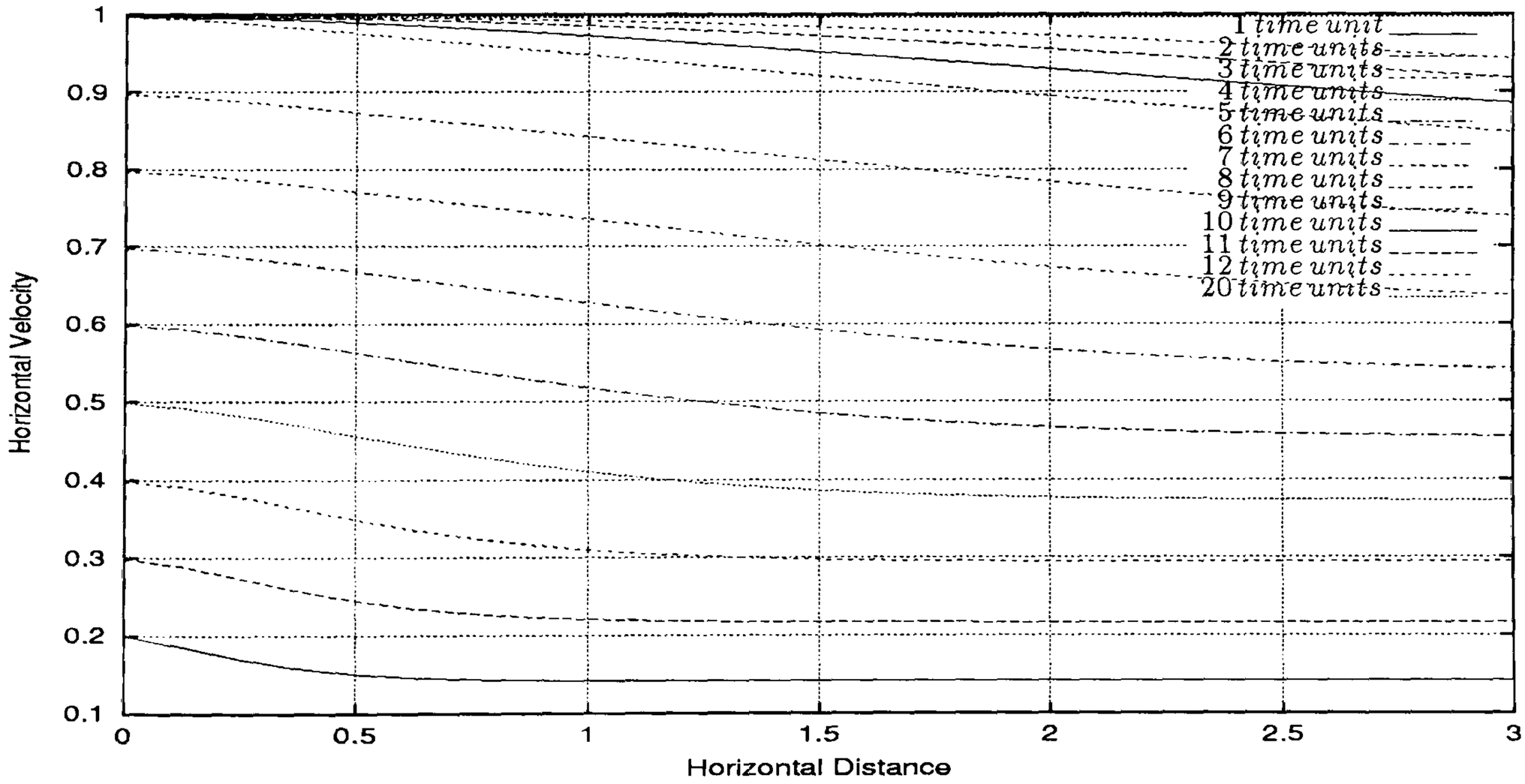


Figure 5.16: Single-phase channel flow - horizontal velocity along the line $y = 0.5$ for an accelerating fluid flow. It can be seen the inlet acceleration from 0.2 to 1.0 and the horizontal parallel curves over all the horizontal distance.

5.2.2 Single-Phase Flow Through an Expansion

Problem definition

In this test case, a flow of a single-phase fluid through an expansion is considered. This is a much more complicated case. The geometry of this problem is much more complex. The existence of a narrow region in the domain originates that the variations on the horizontal velocity are expected to be bigger than in a straight channel flow. The vertical velocity also has important changes before and right after the geometric constraint. Furthermore the horizontal velocities can vary from positive values to negative values in the region just after the constraint. Thus in this recirculation region the differential equations which represent the flow will be harder to solve. Naturally this recirculation region will depend on the Reynolds number of the fluid flow. Hence the physical properties chosen are:

Flow Regime	Reynolds Number	Viscosity	Density
Single-Phase	100	0.0125	1.0

where the Reynolds number is defined like in the Channel Flow test case (see (Equation 5.1)).

Figure 5.17 shows the computational domain chosen for this simulation. It is defined by

$$\begin{cases} 0 \leq x/(2h) \leq 3 & for \quad 0 \leq y/(2h) \leq 0.5 \\ 3 \leq x/(2h) \leq 15 & for \quad -0.5 \leq y/(2h) \leq 1.0 \end{cases}$$

The resolution of the coarsest grid is $\Delta_{xsize} = 0.1875$ and $\Delta_{ysize} = 0.125$. In Figure 5.18 we see a grid at the uniform level 3. Figure 5.19 shows the sections where the solution profiles are going to be presented.

The boundary conditions used are at the inlet, Dirichlet boundary conditions and at the outlet Neumann conditions. In the inlet a parabolic velocity profile is specified for the x-component velocity $u(x, y)$ by

$$u(0, y) = 4y(0.5 - y)$$

and for the y-component velocity $v(x, y)$ it is null in the inlet boundary ($v(0, y) = 0$).

The initial condition for the simulation is exactly the same as in the channel test case. The velocities are all set to zero. The convergence tolerance for the simulation is 10^{-6} for each time step.

Results for a 3 level computation

Figure 5.20 shows the streamlines computed for this test case, after 20 time units.

In Figures 5.21 and 5.22 we present the horizontal and vertical velocity profiles along the line $x = 3.375$ for a 3 level computation. This region is a very difficult

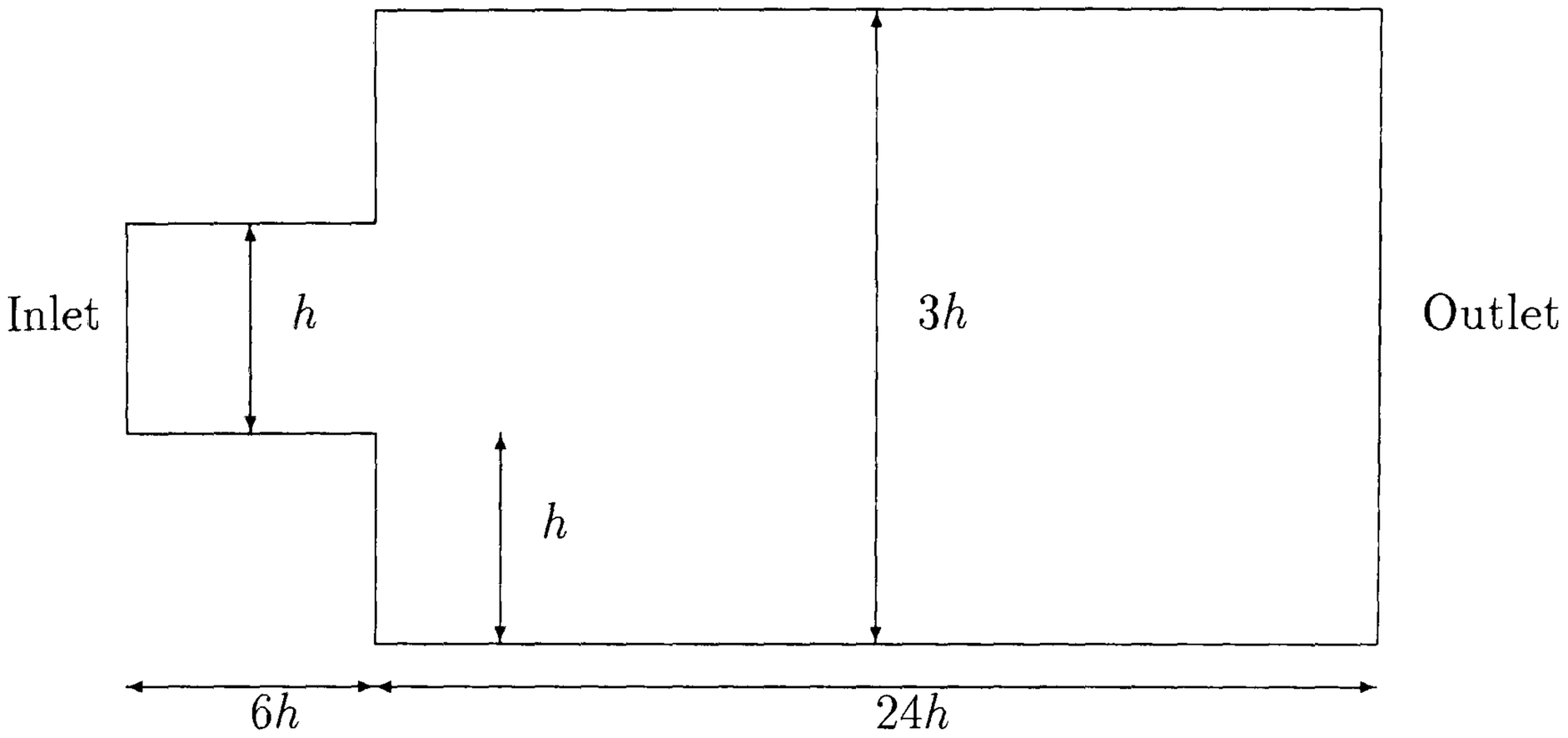


Figure 5.17: Geometrical representation of the domain of the Expansion flow problem

one since the solution is varying significantly just after the step. It is visible the different signs of the vertical velocity over the vertical distance.

Figure 5.23 show the horizontal velocity profiles along the line $x = 14.375$ for a 3 level computation. In this region, which is very close to the outlet, the horizontal velocities exhibit a parabolic profile. Furthermore the bigger velocity is about 0.3 of the maximum value in the inlet. This is precisely what we are expecting since the relation between the height in the inlet and the height in the outlet is precisely 0.3. The vertical velocity component is zero. Figure 5.24 presents the pressure drop profile along the line $y = 0.25$ for a 3 level computation. It is visible that even after 10 and 20 time units the pressure drop agree very well and exhibit the same pattern. Two different straight lines each one correspondent to the two regions of the domain with different height.

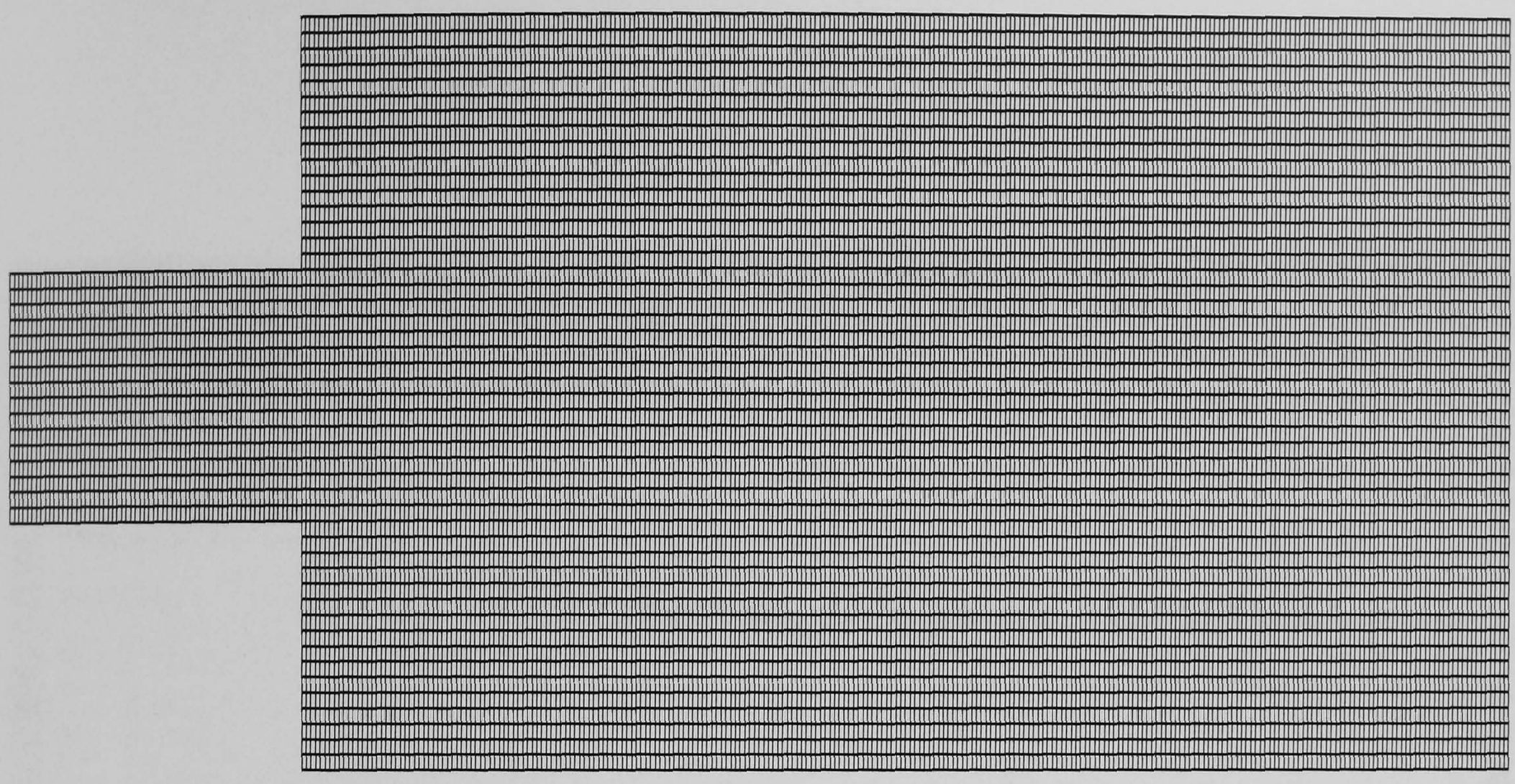


Figure 5.18: Uniform level 3 grid for the Expansion flow test case ($\Delta_{xsize} = 0.046875$ and $\Delta_{ysize} = 0.03125$)

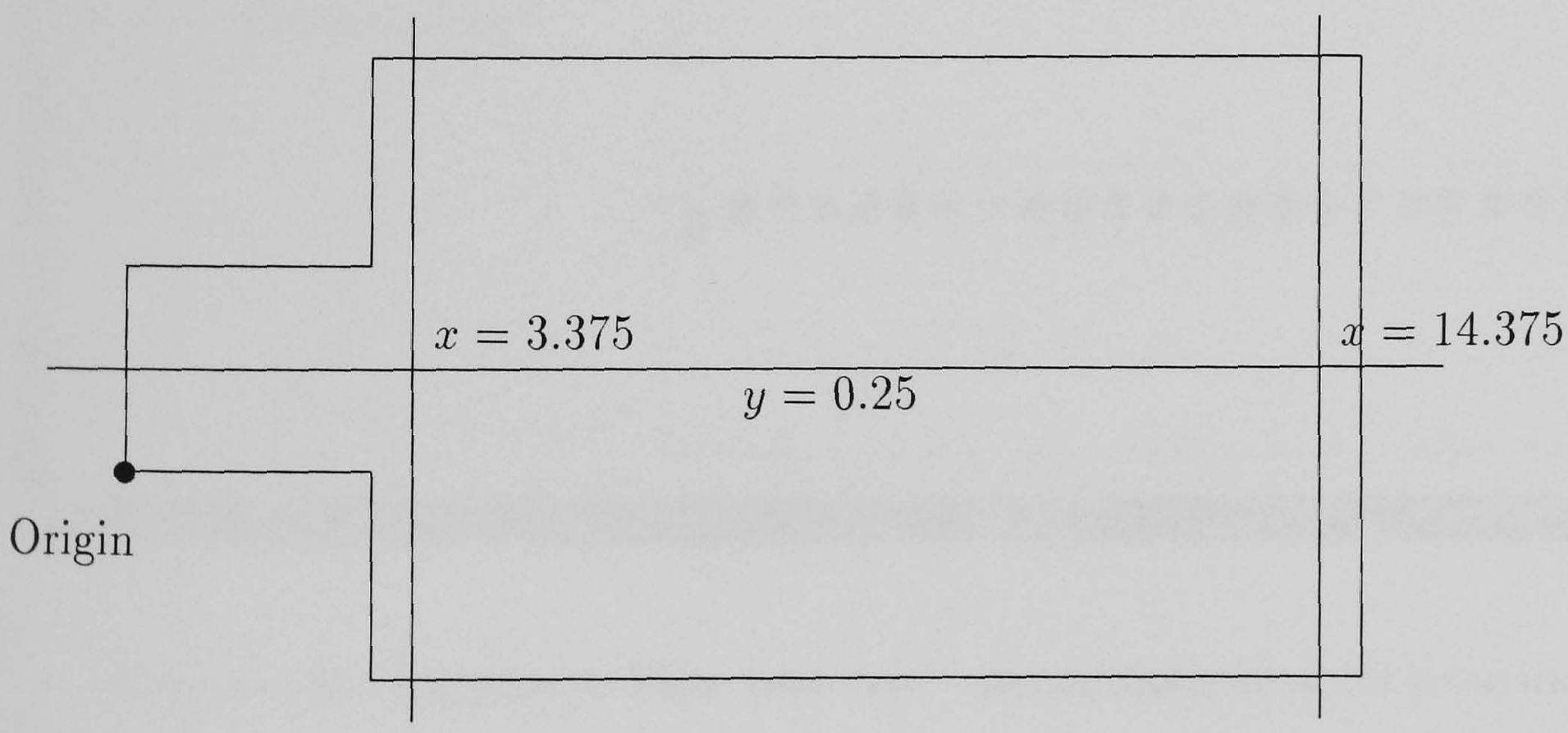


Figure 5.19: Expansion Flow test case - main sections used to give the solution profiles

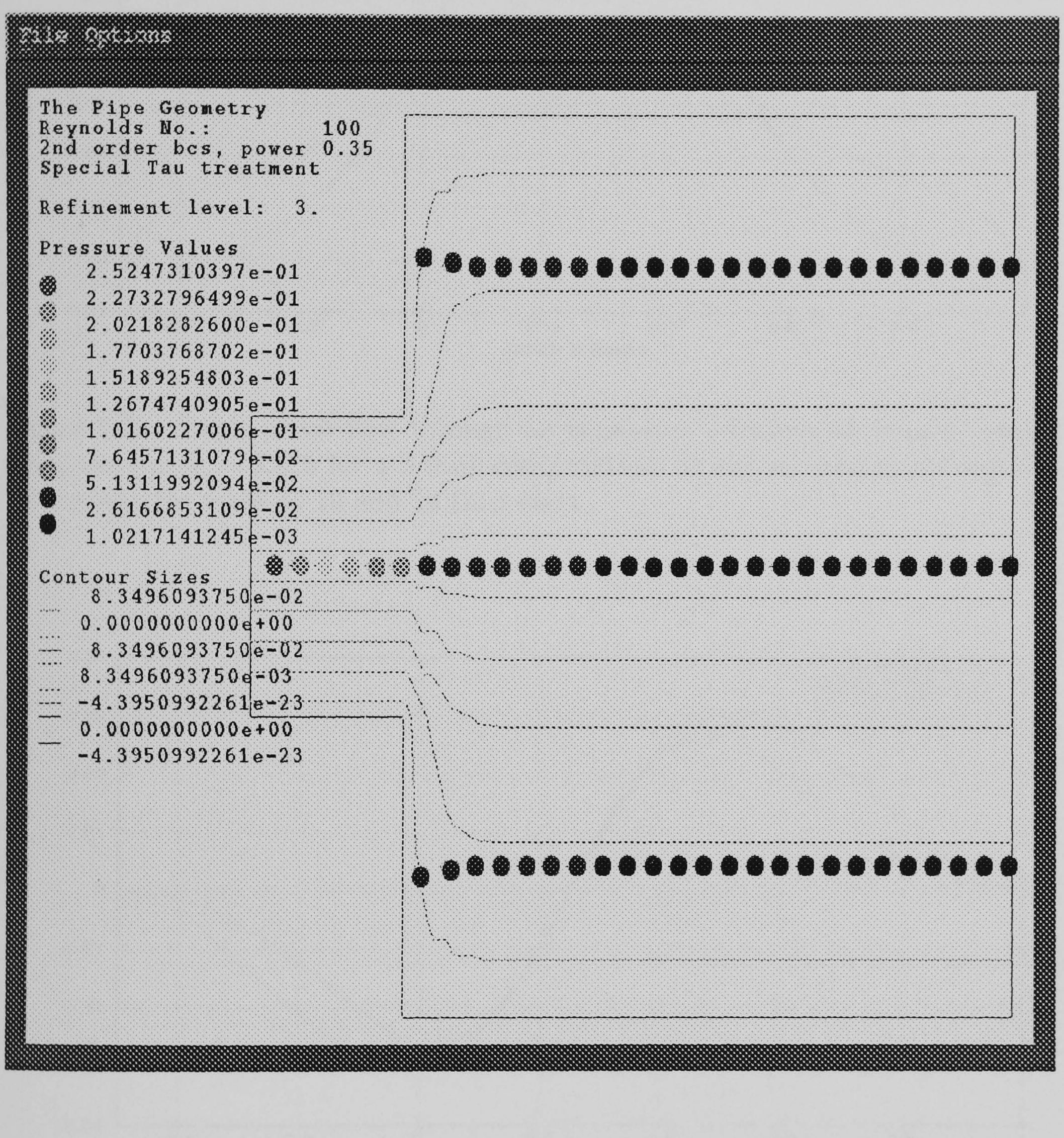


Figure 5.20: Expansion Flow test case - streamlines after 20 time units

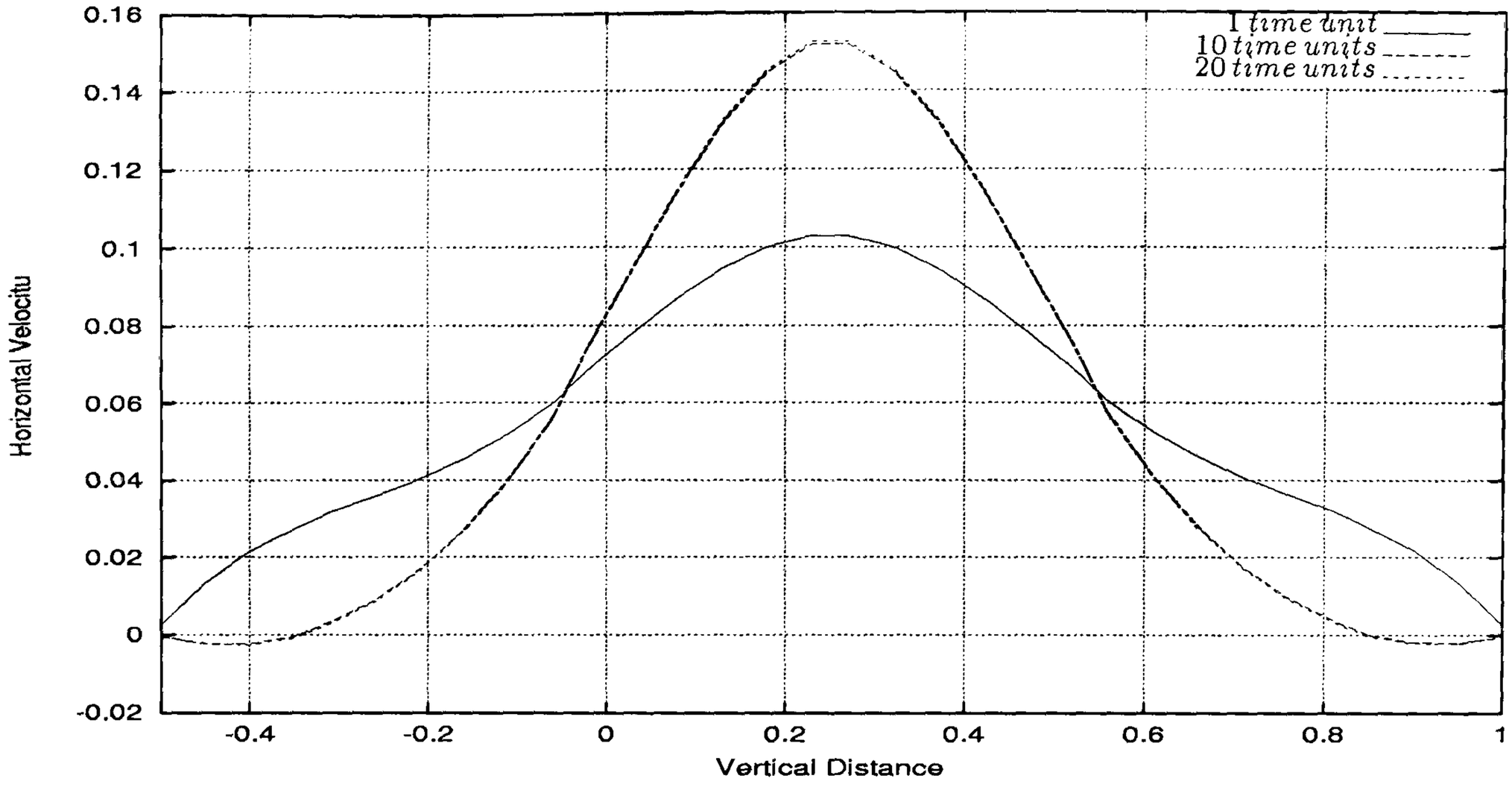


Figure 5.21: Single-phase flow through an expansion - horizontal velocity profile along the line $x = 3.375$ for a 3 level computation. It can be seen small negative values near the walls at 10 and 20 time units.

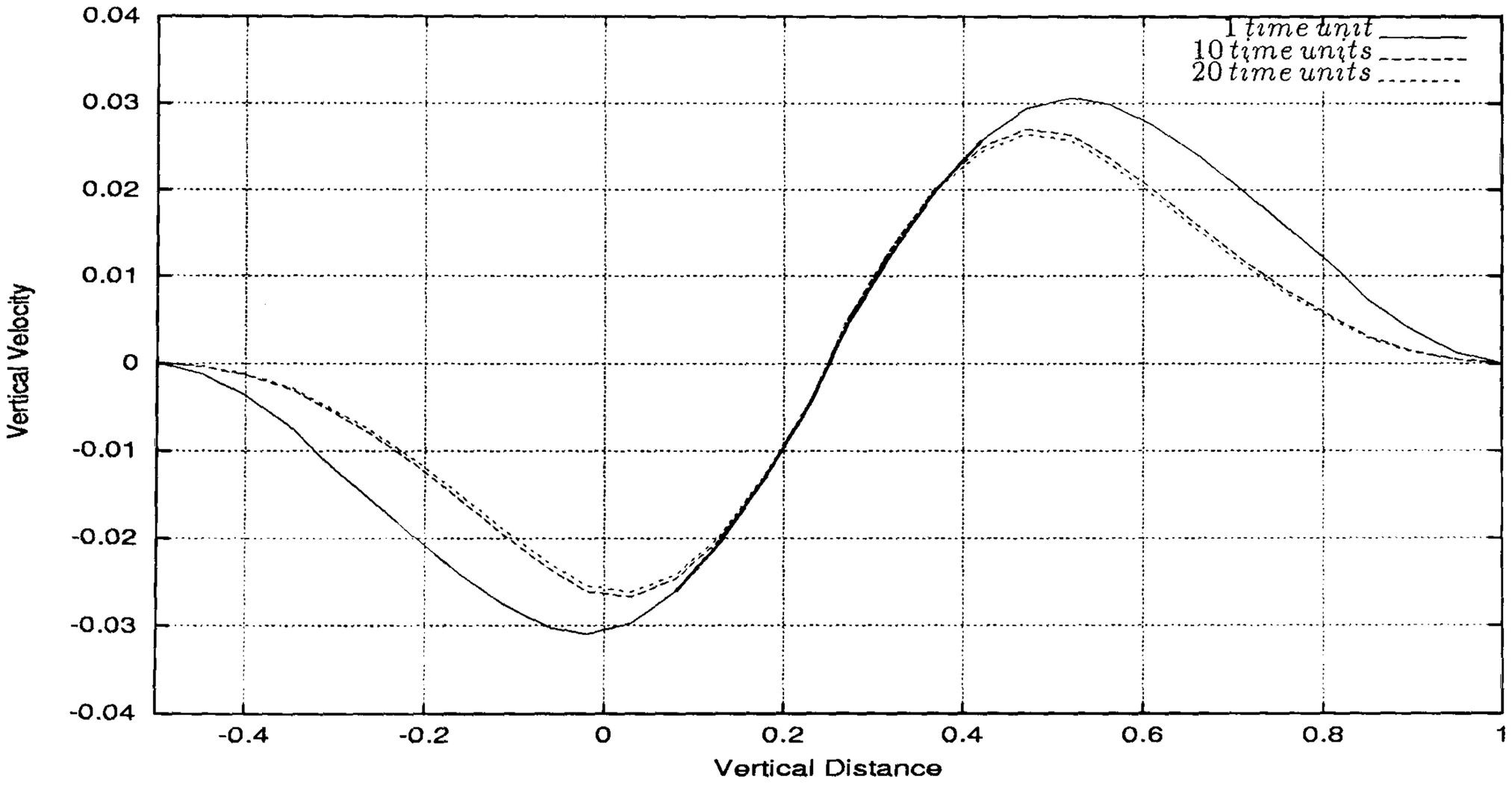


Figure 5.22: Single-phase flow through an expansion - vertical velocity profile along the line $x = 3.375$ for a 3 level computation. It can be observed the different signs of the vertical velocity and the null values in the walls and in the middle of the vertical distance.

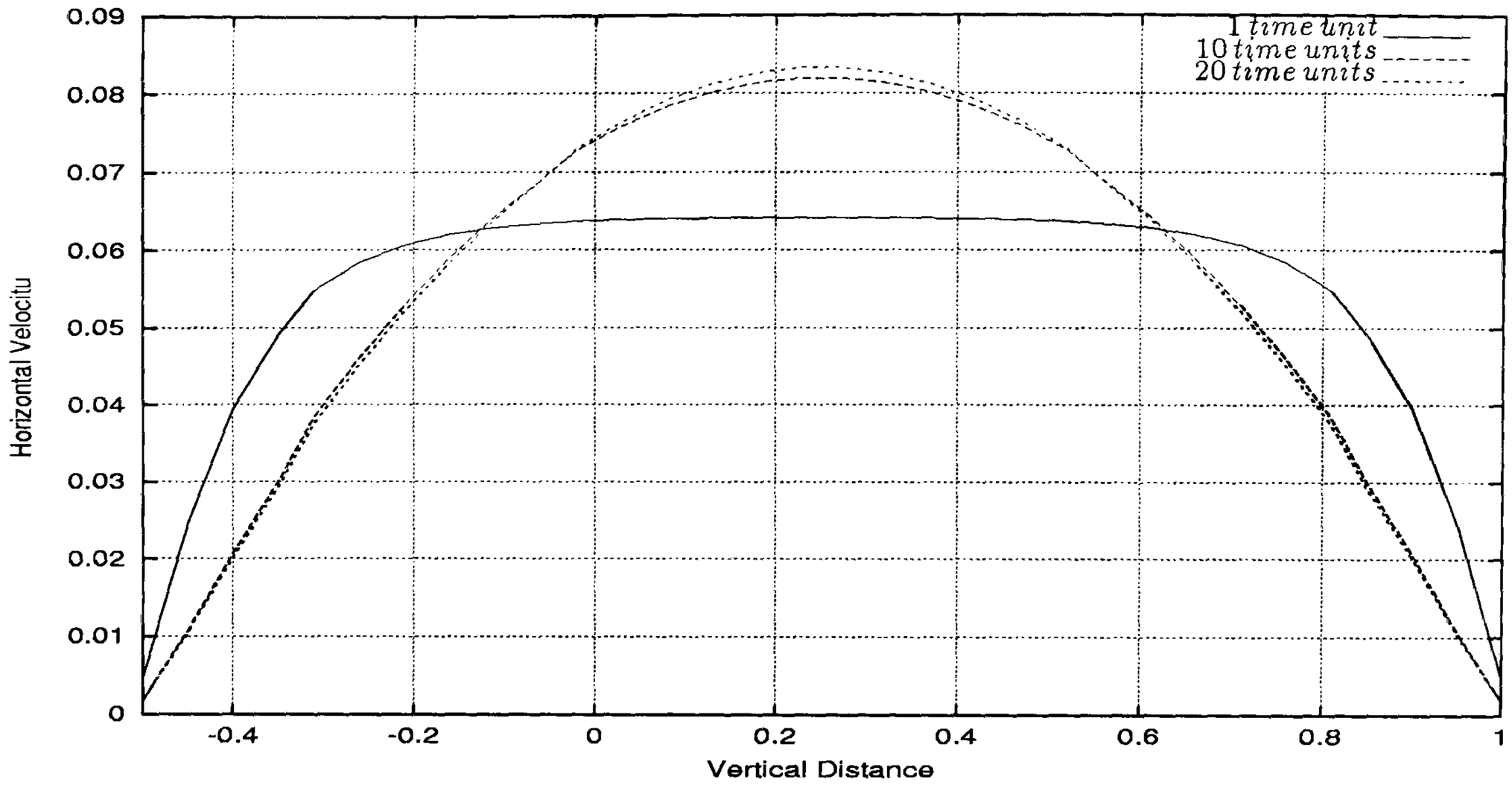


Figure 5.23: Single-phase flow through an expansion - horizontal velocity profile along the line $x = 14.375$ for a 3 level computation. It can be seen the fully developed parabolic profile at 10 and 20 time units.

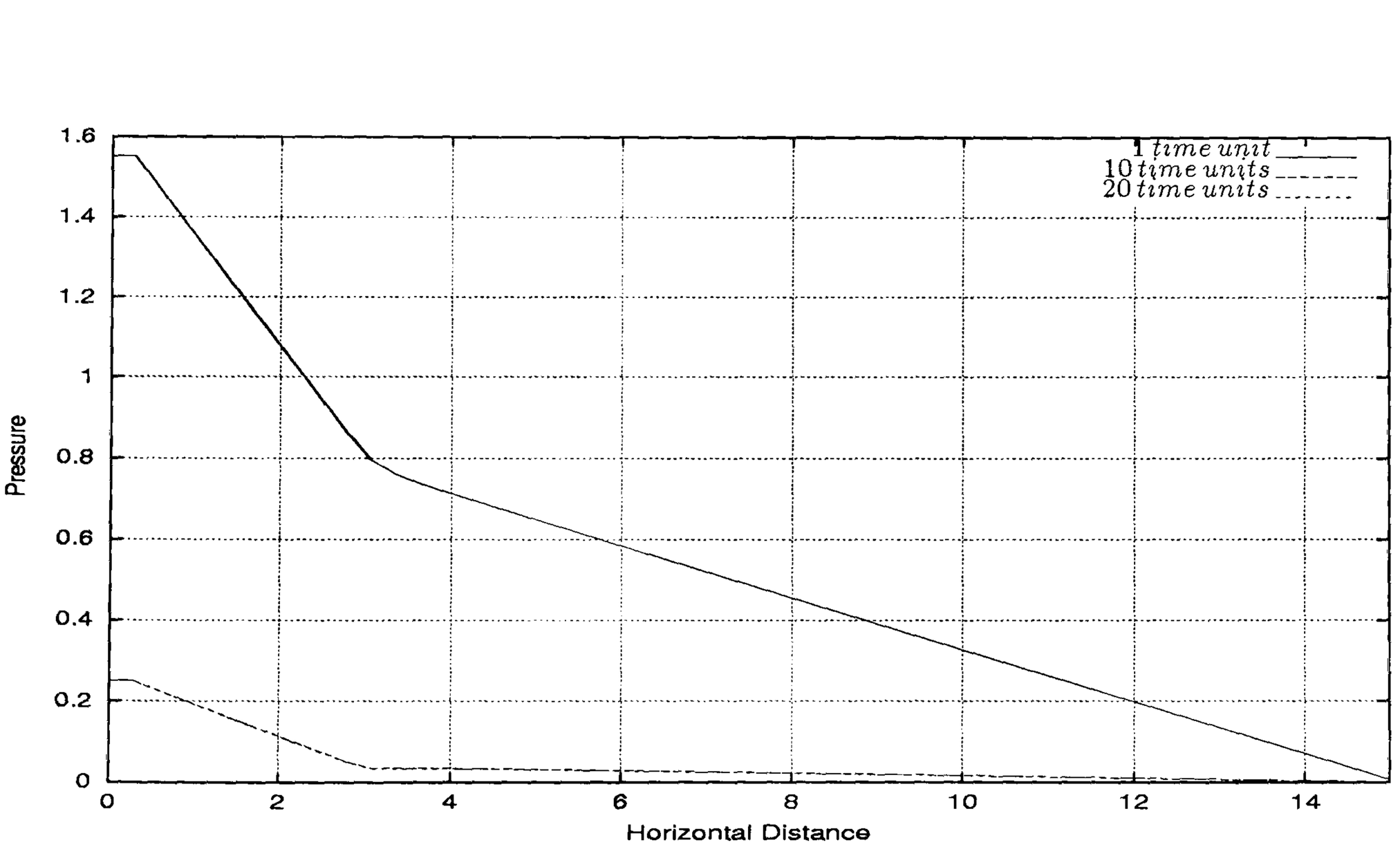


Figure 5.24: Single-phase flow through an expansion - pressure drop profile along the line $y = 0.25$ for a 3 level computation. It can be seen good agreement between the pressure drop profiles after 10 and 20 time units.

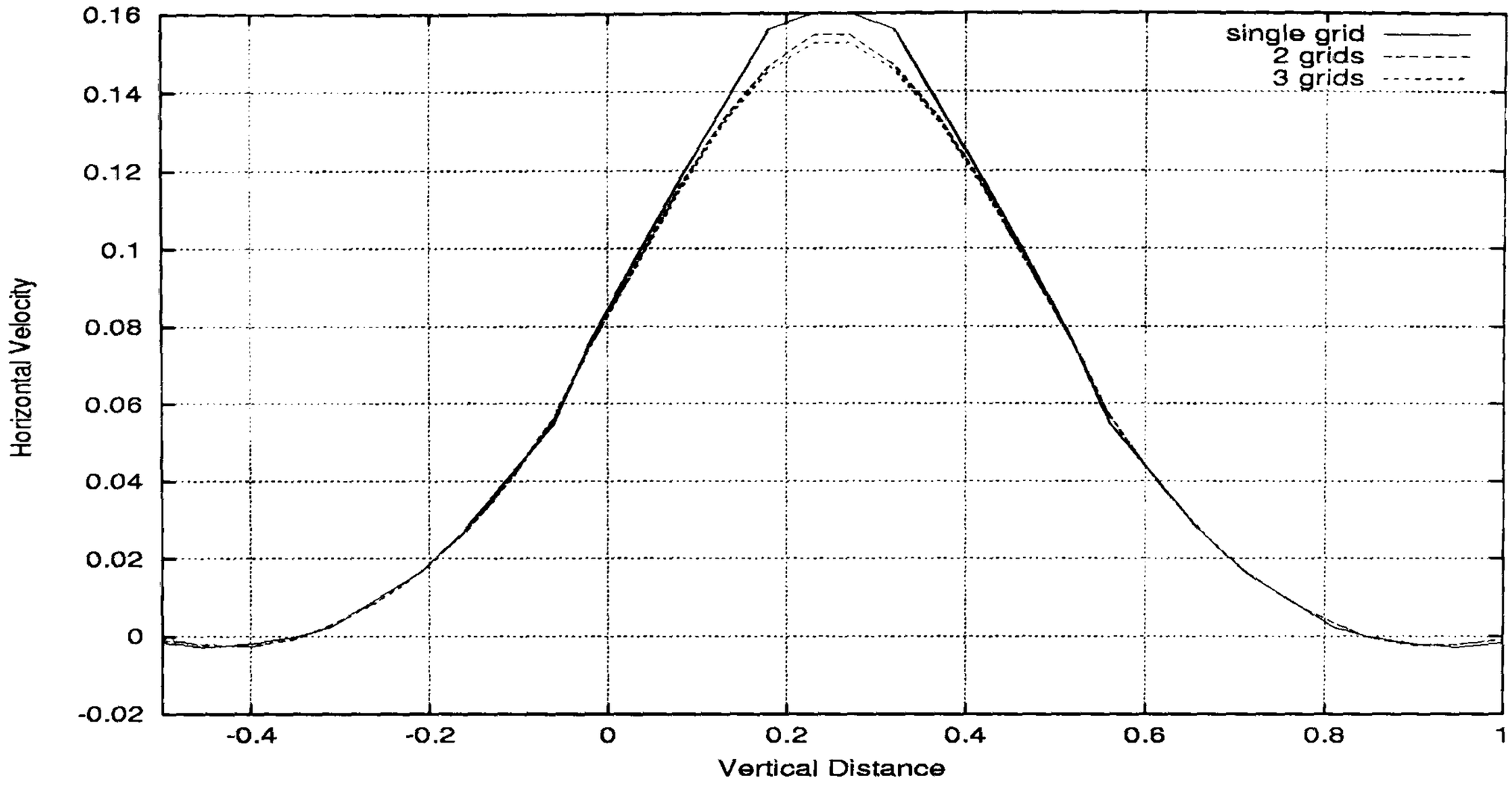


Figure 5.25: Single-phase flow through an expansion - horizontal velocity profile along the line $x = 3.375$ for single 2 and 3 level computation at 20 time units. It can be observed the better parabolic profiles developed for 2 and 3 level calculations.

Grid and Time Step Independence of the Solutions

Figures 5.25 5.26 and 5.27 represent the horizontal and vertical velocity profiles for single 2 and 3 level computations at 20 time units. Good agreement is observed between the profiles. It can be also observed in Figures 5.25 and 5.27 that the best parabolic profile is the correspondent to the 3 level calculation. This is exactly what it should be expected because in a 3 level calculation we have a bigger number of cells than in a 2 level or single grid calculation. Consequently the discretization gives a better approximation of the differential equations. The vertical velocity along the line $x = 14.375$ is null for all the grid cases studied.

In Figures 5.28 through 5.31 profiles for different time steps are presented. Good agreement that can be observed between the time step 1.0 and 10.0 profiles, particularly at 20 time units. This suggests that the solver is independent on the time step size.

Comparison with `CFX 4.3` Solutions

Figures 5.32 and 5.33 show the horizontal and vertical velocity profiles for `transient Pamg Single-phase` and `CFX 4.3` along the line $x = 3.375$ at 20 time units. In this difficult region, it can be observed the almost perfect agreement between the solutions produced by both solvers. In Figure 5.34 the profiles of the horizontal velocities along the line $x = 14.375$ at 20 time units are also very close. Figure 5.35 shows that for the `transient Pamg Single-phase` the vertical velocity at $x = 14.375$ is always zero for each point of the vertical distance. Although the `CFX 4.3` profile is different, the values of the vertical velocity are very small and consequently we can establish that the vertical velocity along the line $x = 14.375$

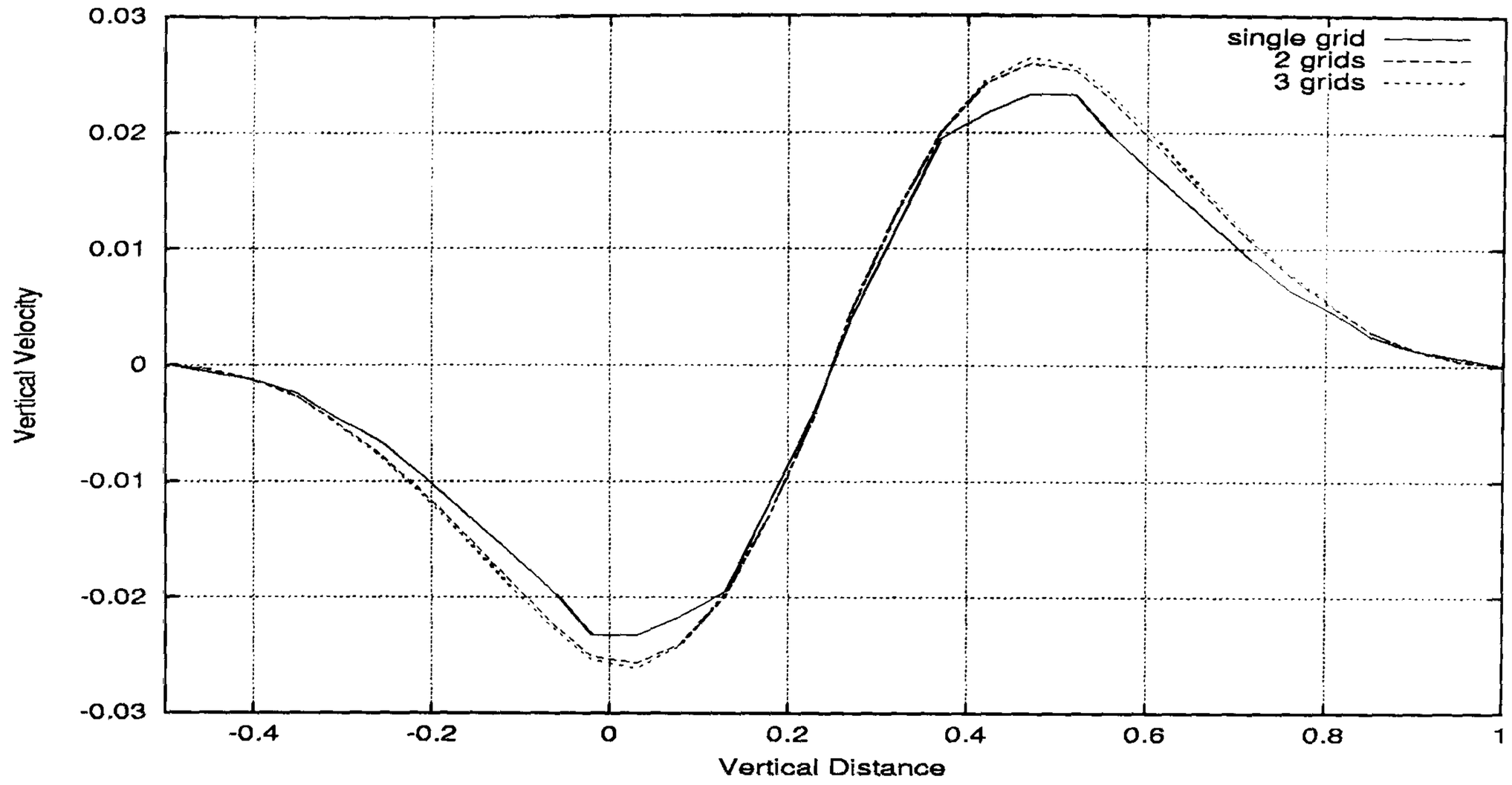


Figure 5.26: Single-phase flow through an expansion - vertical velocity profile along the line $x = 3.375$ for a single 2 and 3 level computation at 20 time units. It can be seen the better developed shape for 2 and 3 level calculations.

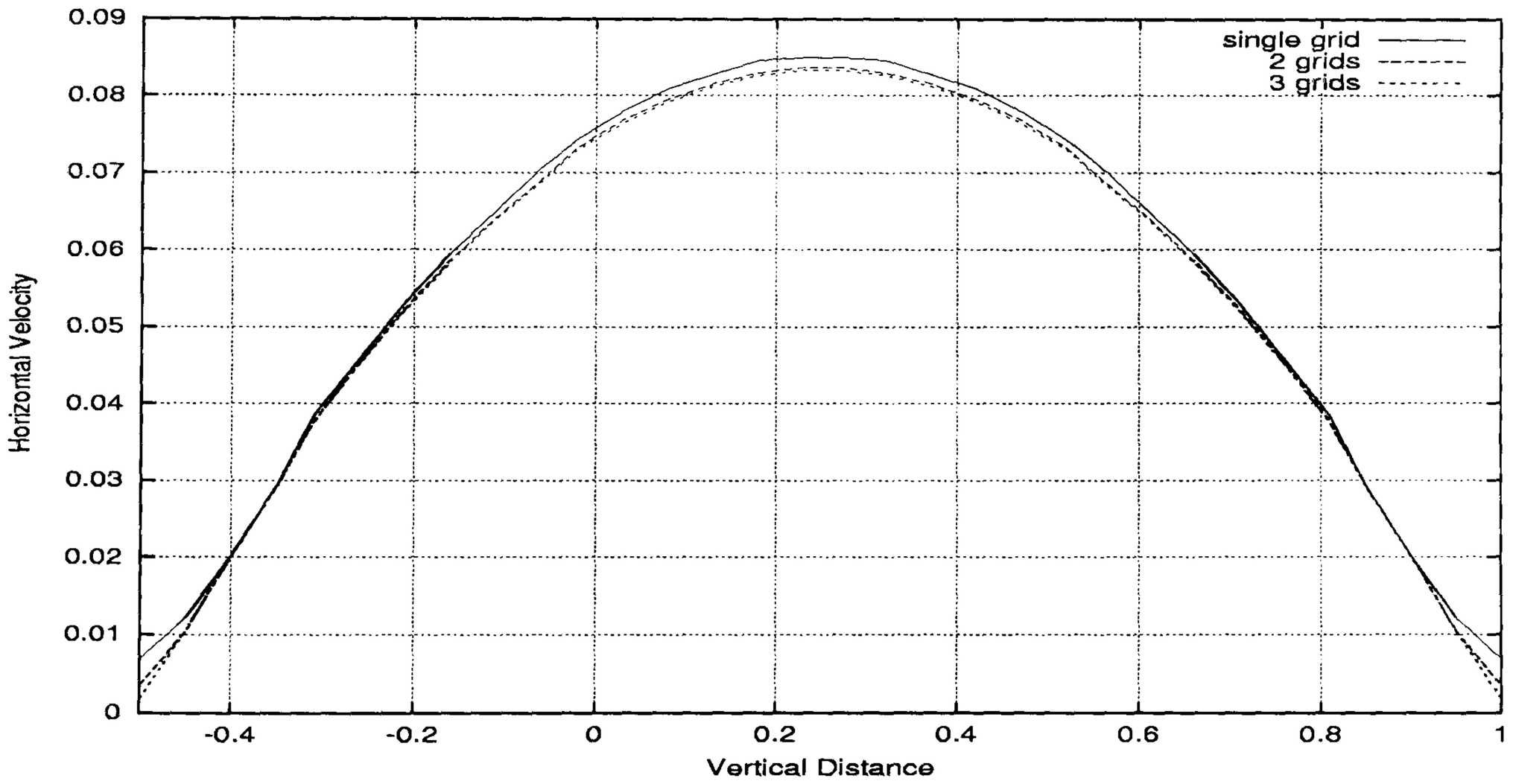


Figure 5.27: Single-phase flow through an expansion - horizontal velocity along the line $x = 14.375$ for single 2 and 3 level computation at 20 time units. It can be observed the fully developed parabolic profile for all level calculations.

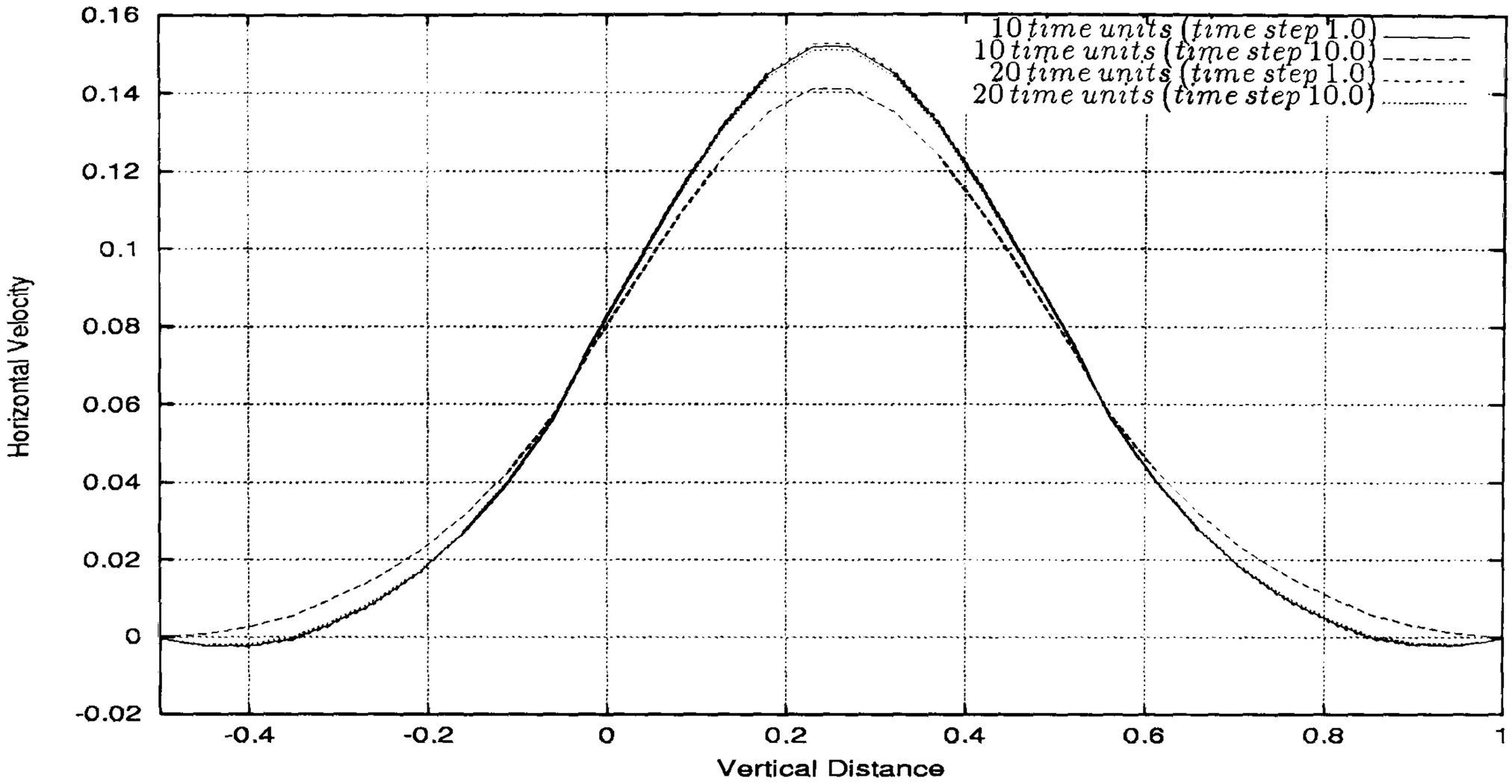


Figure 5.28: Single-phase flow through an expansion - horizontal velocity profile along the line $x = 3.375$ for 3 level computation for different time steps. It can be seen the good agreement between the parabolic profiles at 20 time units.

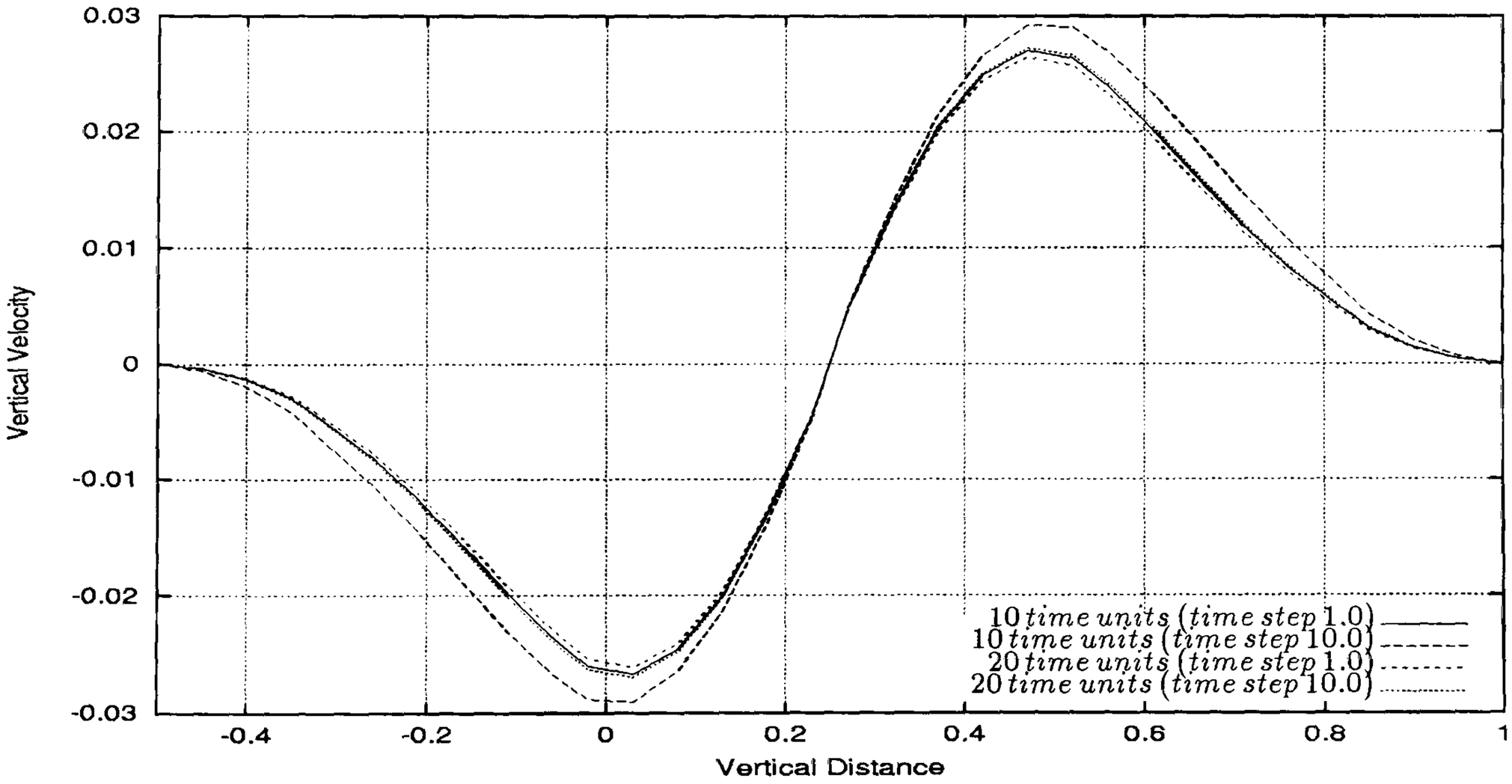


Figure 5.29: Single-phase flow through an expansion - vertical velocity profile along the line $x = 3.375$ for 3 level computation for different time steps. It is visible the good agreement between the profiles at 20 time units.

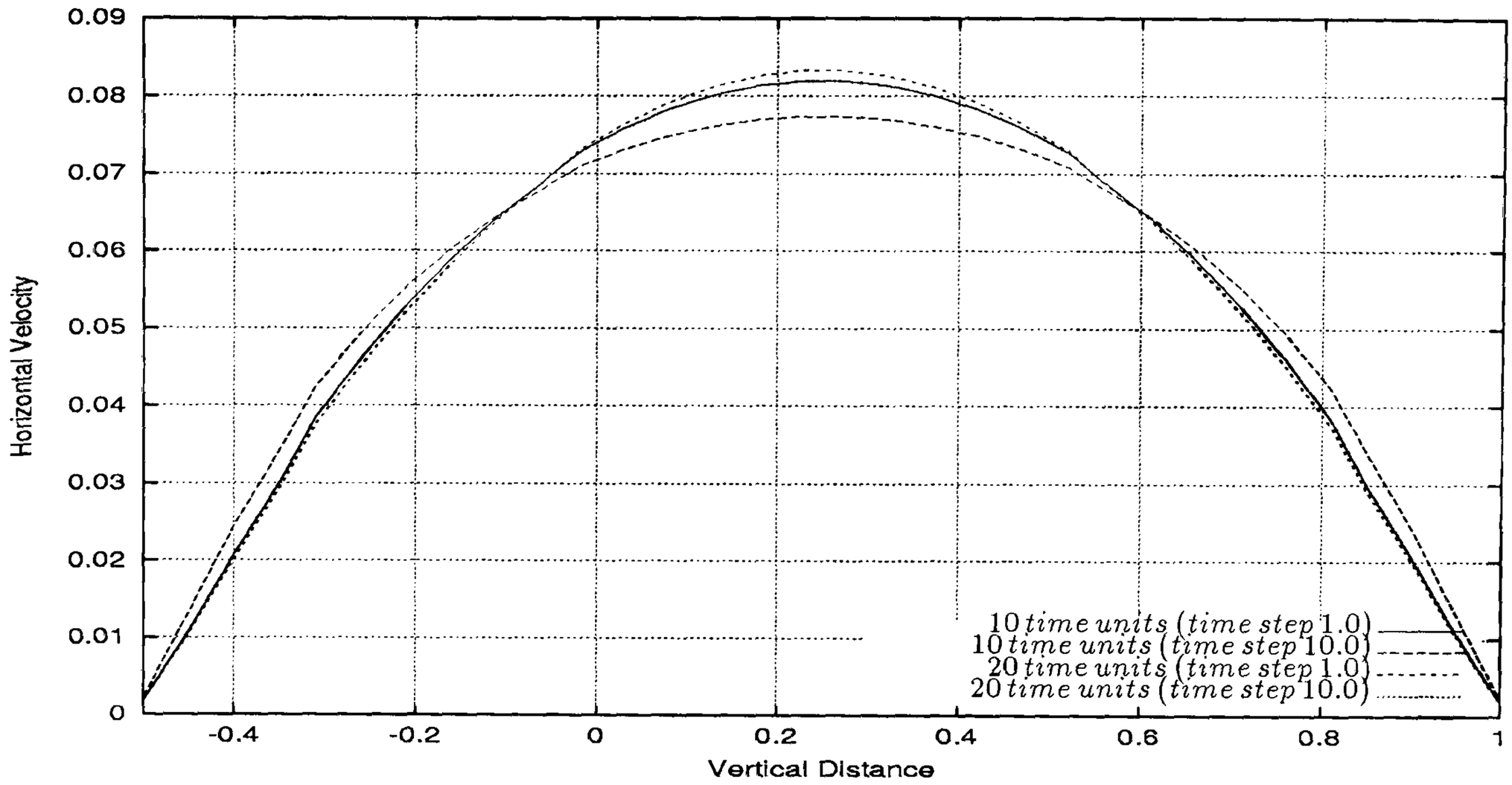


Figure 5.30: Single-phase flow through an expansion - horizontal velocity profile along the line $x = 14.375$ for 3 level computation for different time steps. It can be seen the good agreement between the parabolic profiles at 20 time units.

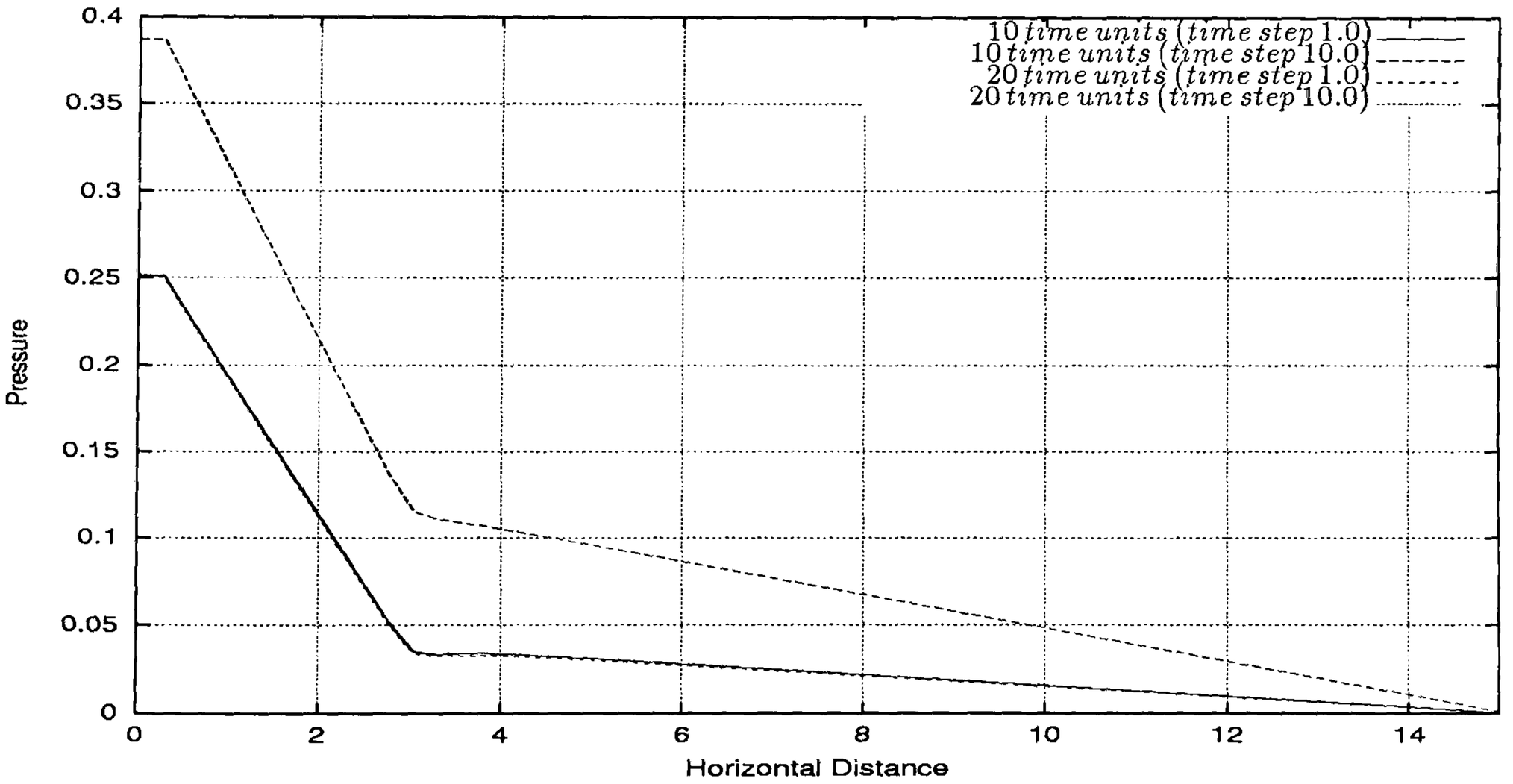


Figure 5.31: Single-phase flow through an expansion - pressure drop profile along the line $y = 0.25$ for 3 level computation for different time steps. It can be observed the same pattern for all the pressure drop curves.

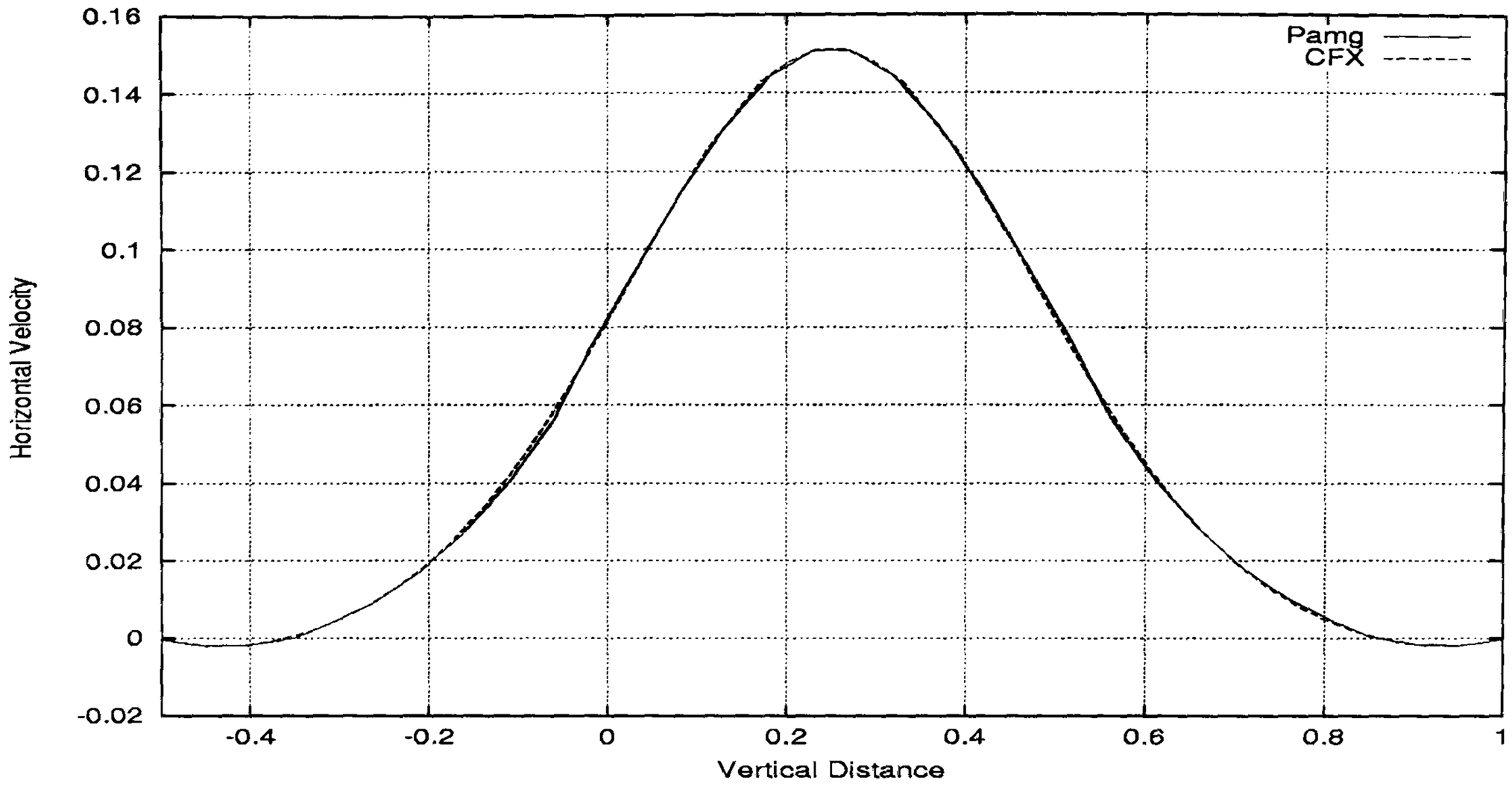


Figure 5.32: Single-phase flow through an expansion - horizontal velocity profile along the line $x = 3.375$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be seen good agreement between the profiles produced by both solvers.

tend to zero. Finally in Figure 5.36 the pressure drop profiles are presented. The values of the pressure drop giving by `CFX 4.3` and `transient Pamg Single-Phase` are very close. In the inlet it can be observed a slight variation which results from how the pressure is computed in the inlet. `transient Pamg Single-phase` equalizes the value in the inlet to the value of the first cell, while `CFX 4.3` extrapolates the value computed in the first cell to the inlet. In both solvers the pressure is computed in the centre of the cells.

Summary

This expansion test case has been chosen in order to study the behaviour of the `transient Pamg Single-phase` in a reentry fluid flow. Hence we could verify how the solver gives the correct solutions in a more complex domain than channel case. It has been verified in this experiment that the `transient Pamg Single-phase` depends on the Reynolds number.

The results presented for Reynolds number 100 show a very good agreement with the same results produced by the `CFX 4.3` solver. Furthermore this confirms that the implementation is correct.

Like in the channel case some tests were undertaken to verify the grid independence and time step independence of the multigrid algorithm. Good agreement could be verified for single 2 and 3 level calculations and for calculations with 1.0 and 10.0 as two different time steps. This reinforces the grid independence and time step independence of the solver, which have been already verified in the Channel Flow test case.

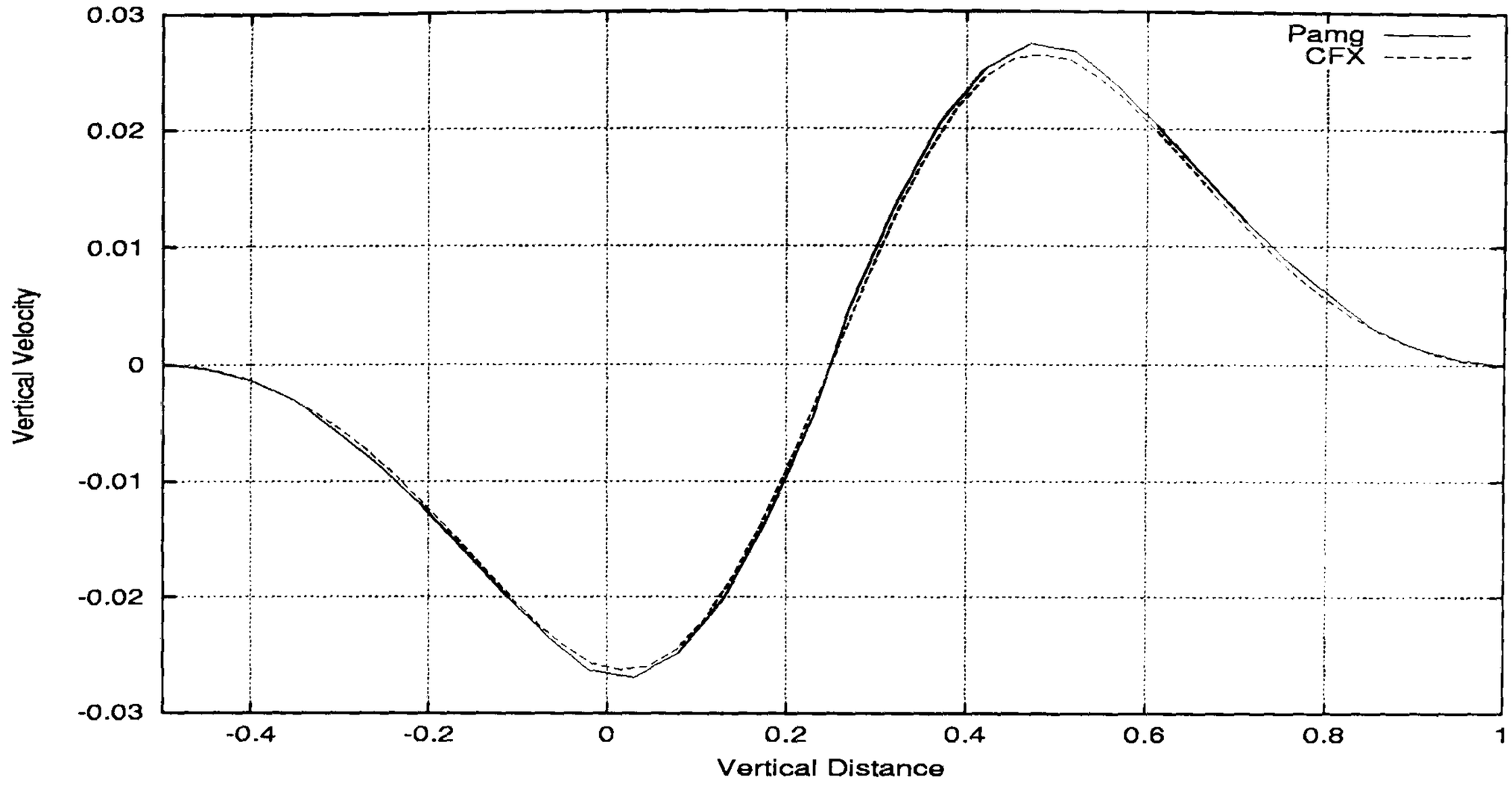


Figure 5.33: Single-phase flow through an expansion - vertical velocity profile along the line $x = 3.375$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. Good agreement can be observed in the vertical velocities produced by both solvers.

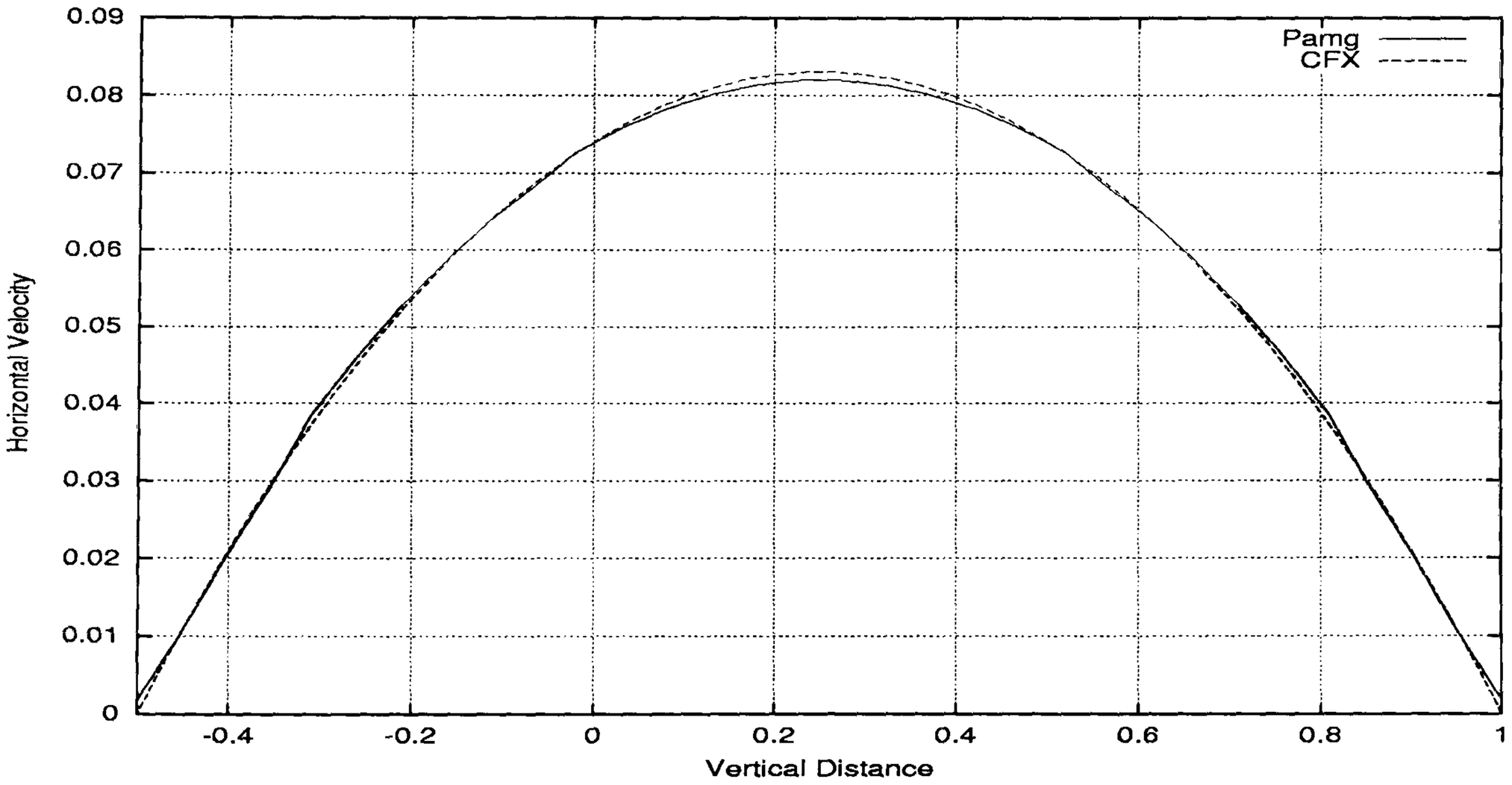


Figure 5.34: Single-phase flow through an expansion - horizontal velocity profile along the line $x = 14.375$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be seen good agreement between the fully developed parabolic profiles produced by both solvers.

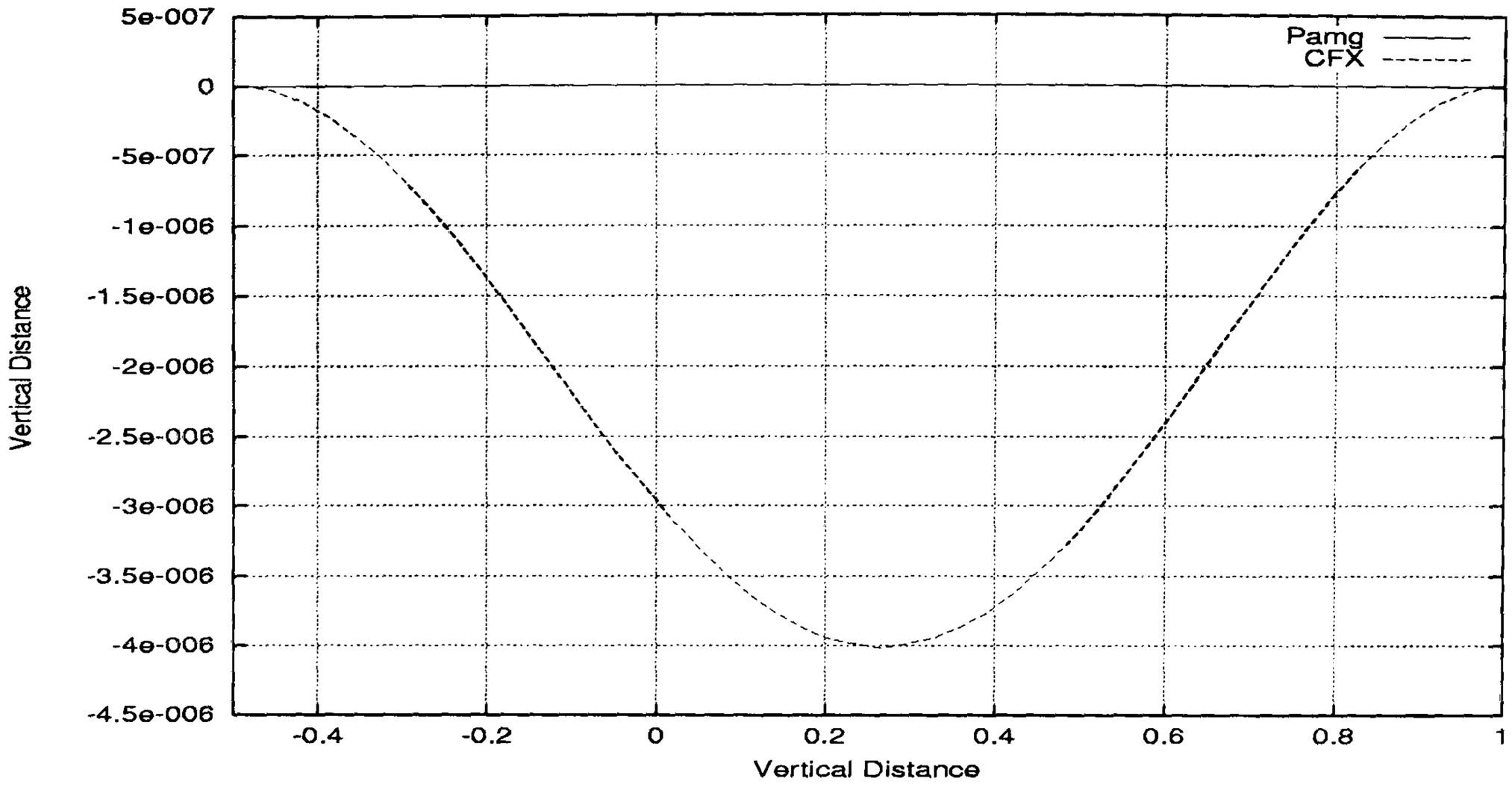


Figure 5.35: Single-phase flow through an expansion - vertical velocity profile along the line $x = 14.375$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be seen that the vertical velocities are close to zero even in the `CFX 4.3` curve. Note the scale of the vertical distance axis.

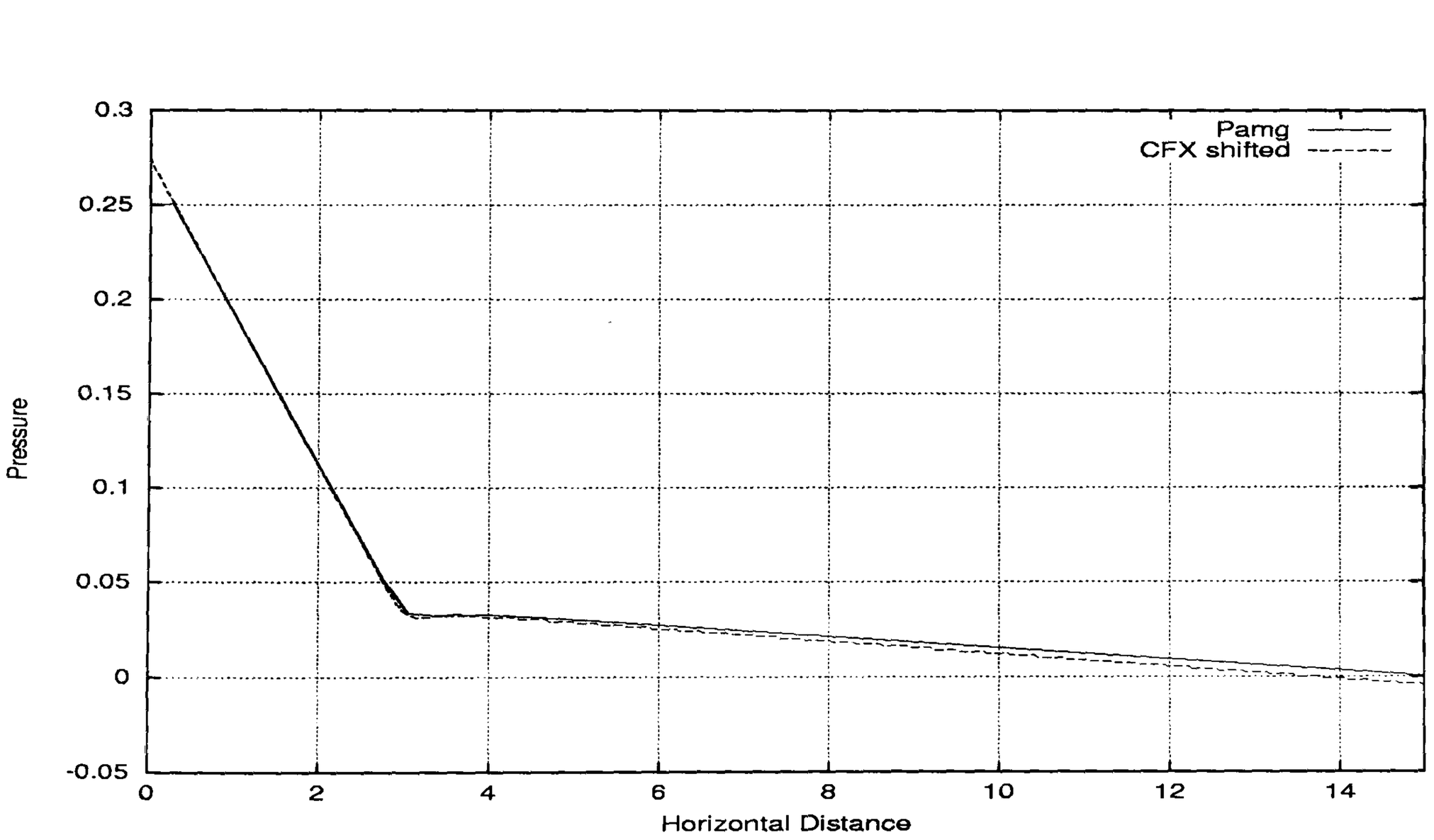


Figure 5.36: Single-phase flow through an expansion - pressure drop profile along the line $y = 0.25$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be observed the good agreement between the pressure drop given by both solvers. Note the slight difference in the inlet.

5.2.3 Single-Phase Flow Through a Contraction

Problem definition

Finally we present the results for the simulation of a single-phase flow through a contraction. This is even a harder test case than the expansion one. In fact in a contraction domain we do not have only the changing of a narrow domain to a larger one. We also have the opposite. Consequently the variations on the fluid behaviour are expected to be bigger. Hence the existence of the contraction region in the middle of the domain is much more demanding for the solution algorithm. Bigger variations in the horizontal velocities can be observed. Furthermore, the vertical component of the velocity changes from positive to negative very quickly, in the neighbourhood of the contraction.

The physical properties chosen for this simulation are:

Flow Regime	Reynolds Number	Viscosity	Density
Single-Phase	100	0.01	1.0

Figure 5.37 shows the computational domain chosen for this simulation. It is defined by

$$\begin{cases} 0 \leq x/(4h/3) \leq 11.25 & 0 \leq y/(4h/3) \leq 1.5 \\ 11.25 \leq x/(4h/3) \leq 12 & 0.5 \leq y/(4h/3) \leq 1.0 \\ 12 \leq x/(4h/3) \leq 23.25 & 0 \leq y/(4h/3) \leq 1.5 \end{cases}$$

The resolution of the coarsest grid is $\Delta_{xsize} = 0.1875$ and $\Delta_{ysize} = 0.125$. Figure 5.38 shows the computational domain for the coarsest grid. Figure 5.39 shows the sections where the solution profiles are going to be presented.

The boundary conditions used are at the inlet, Dirichlet boundary conditions and at the outlet Neumann conditions. In the inlet a parabolic velocity profile is specified for the x-component velocity $u(x, y)$ by

$$u(0, y) = 4y(0.5 - y)$$

and for the y-component velocity $v(x, y)$ it is null in the inlet boundary ($v(0, y) = 0$). At the outlet the y-component velocity is also null.

The initial condition for the simulation is very simple, all the velocity components are set to zero. The convergence tolerance for the simulation is 10^{-6} for each time step.

Results for a 3 level computation

In Figure 5.40 we see the streamlines computed for this test case, after 20 time units. Since the Reynolds number is relatively small it is not expected to see any big recirculation region. Once more the aspect ratio is distorted due to the necessity of the plotting routine in plot the results in a fixed area.

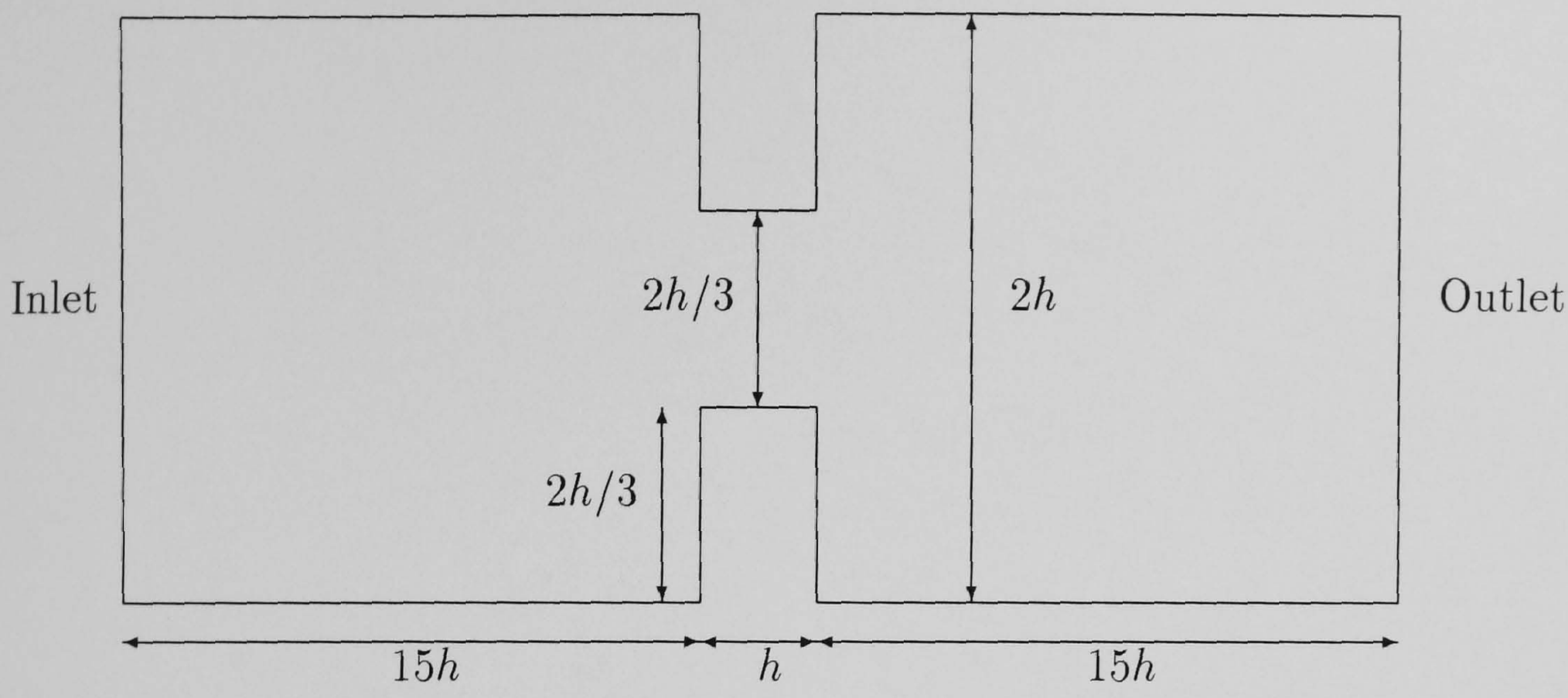


Figure 5.37: Geometrical representation of the domain of the Contraction flow problem

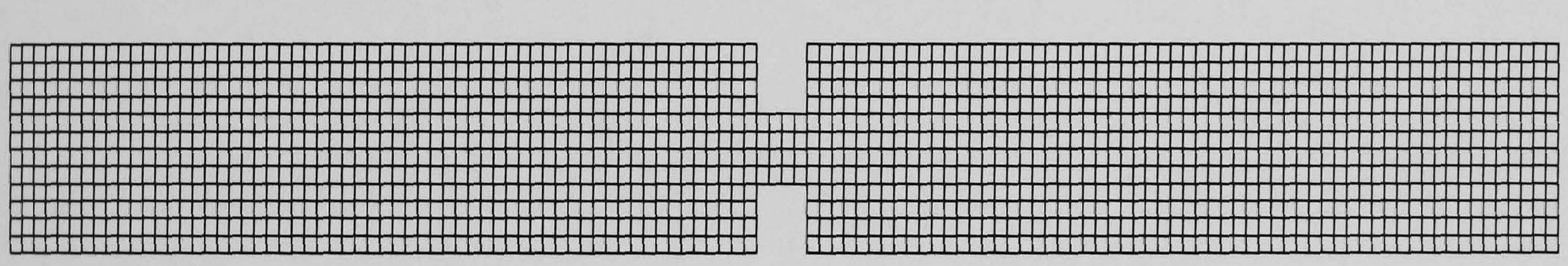


Figure 5.38: Uniform level 1 grid for the Contraction flow test case ($\Delta_{xsize} = 0.1875$ and $\Delta_{ysize} = 0.125$)

Figures 5.41 through 5.44 show for a 3 level computation the profiles for the horizontal velocity and vertical velocity along the lines $x = 10.875$ and $x = 12.375$ at 1, 10 and 20 time units. These profiles are presented in lines, which are situated before and after the contraction region. Figures 5.45 show the horizontal profiles at the outlet. It can be seen the parabolic shape of the horizontal velocity at 10 and 20 time units. The vertical velocity at outlet remains null over all the simulation. In Figure 5.46 the pressure drop along the line $y = 0.75$ is presented. It can be seen the increased pressure drop which occurs in the region of the contraction. The results presented agree with the expectations of the solution profiles. The parabolic shape of the horizontal velocity is observed and the vertical velocity is very small except in the neighbourhood of the contraction. Moreover, the flow accelerates with time and the streamlines do not detach at this small Reynolds Number. The pressure drop is constant over regions of the entire domain.

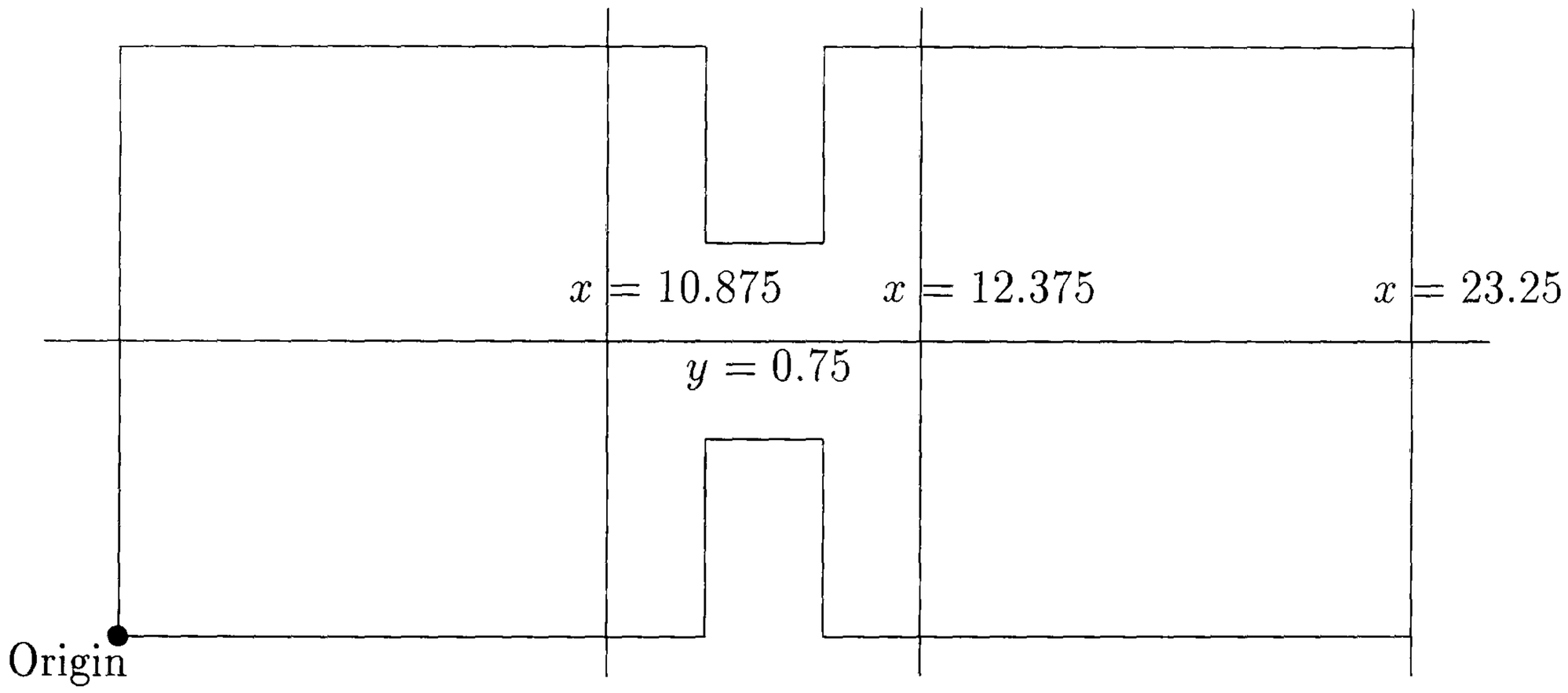


Figure 5.39: Contraction flow test case - main sections used to give the solution profiles

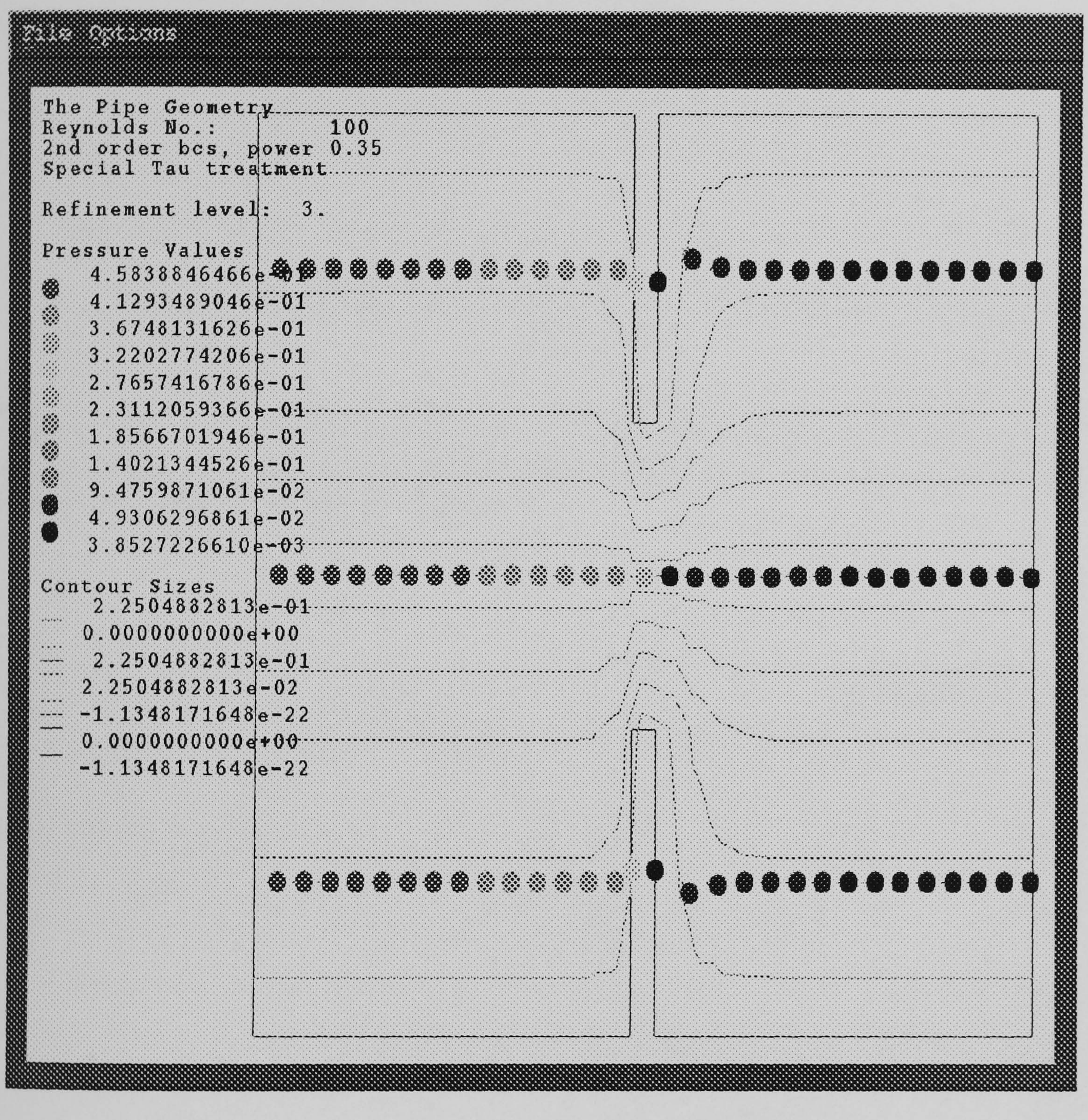


Figure 5.40: Contraction flow test case - streamlines after 20 time units

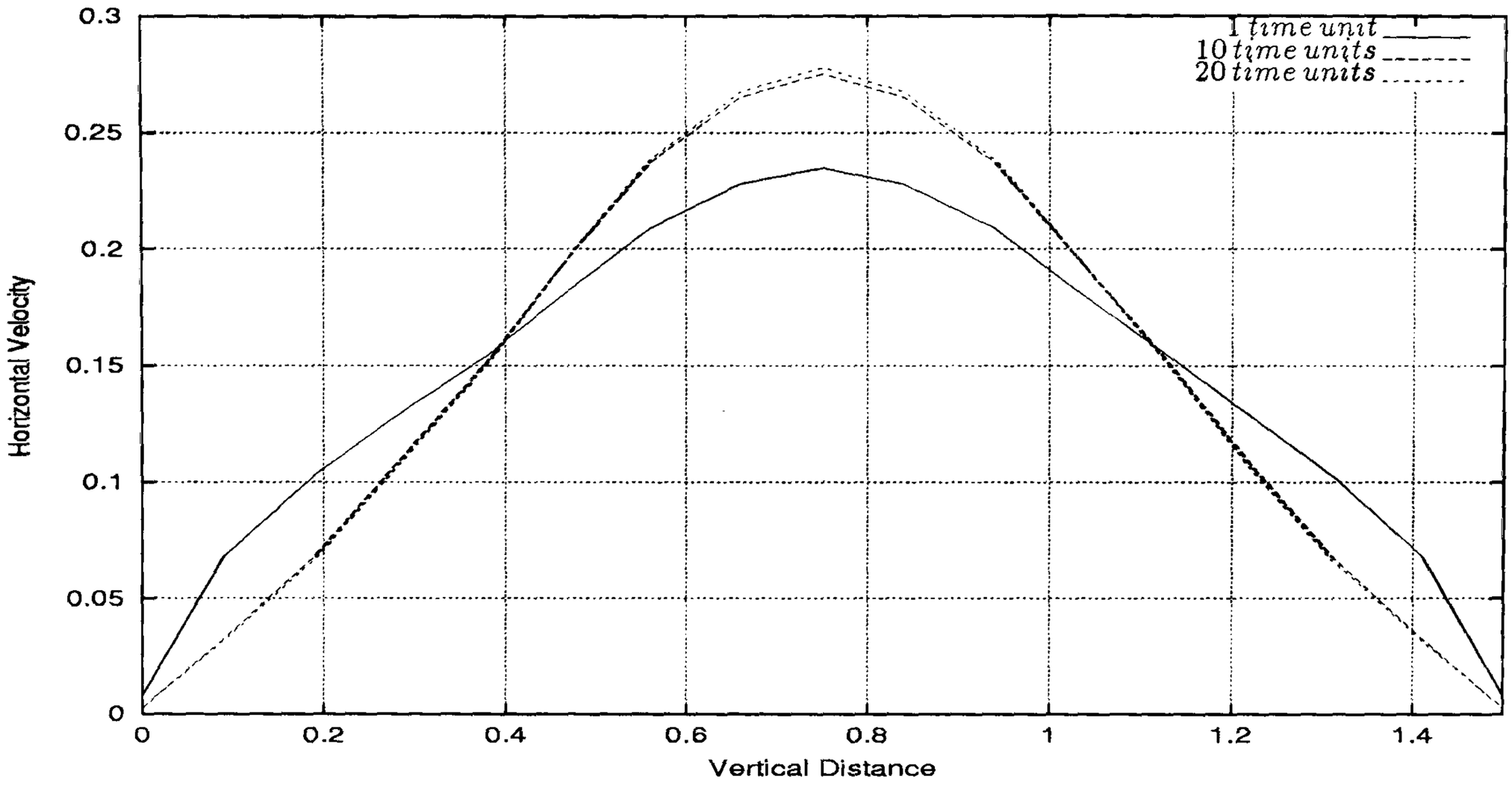


Figure 5.41: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 10.875$ for a 3 level computation. It can be seen the developed parabolic profile at 10 and 20 time units.

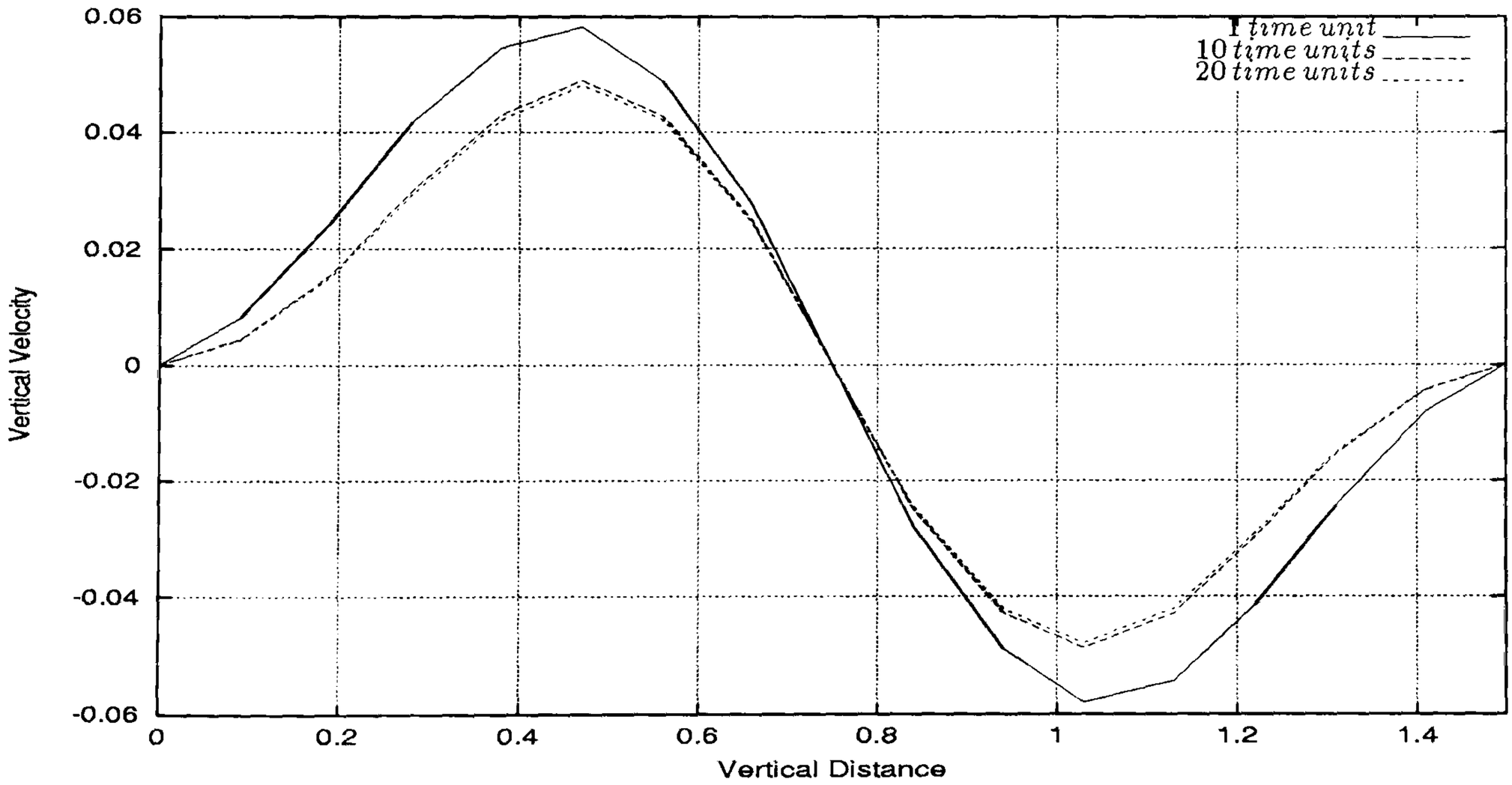


Figure 5.42: Single-phase flow through a contraction - vertical velocity profile along the line $x = 10.875$ for a 3 level computation. It can be observed the positive and negative vertical velocity values over the vertical distance. Note the null values at the walls and in the middle of the vertical distance.

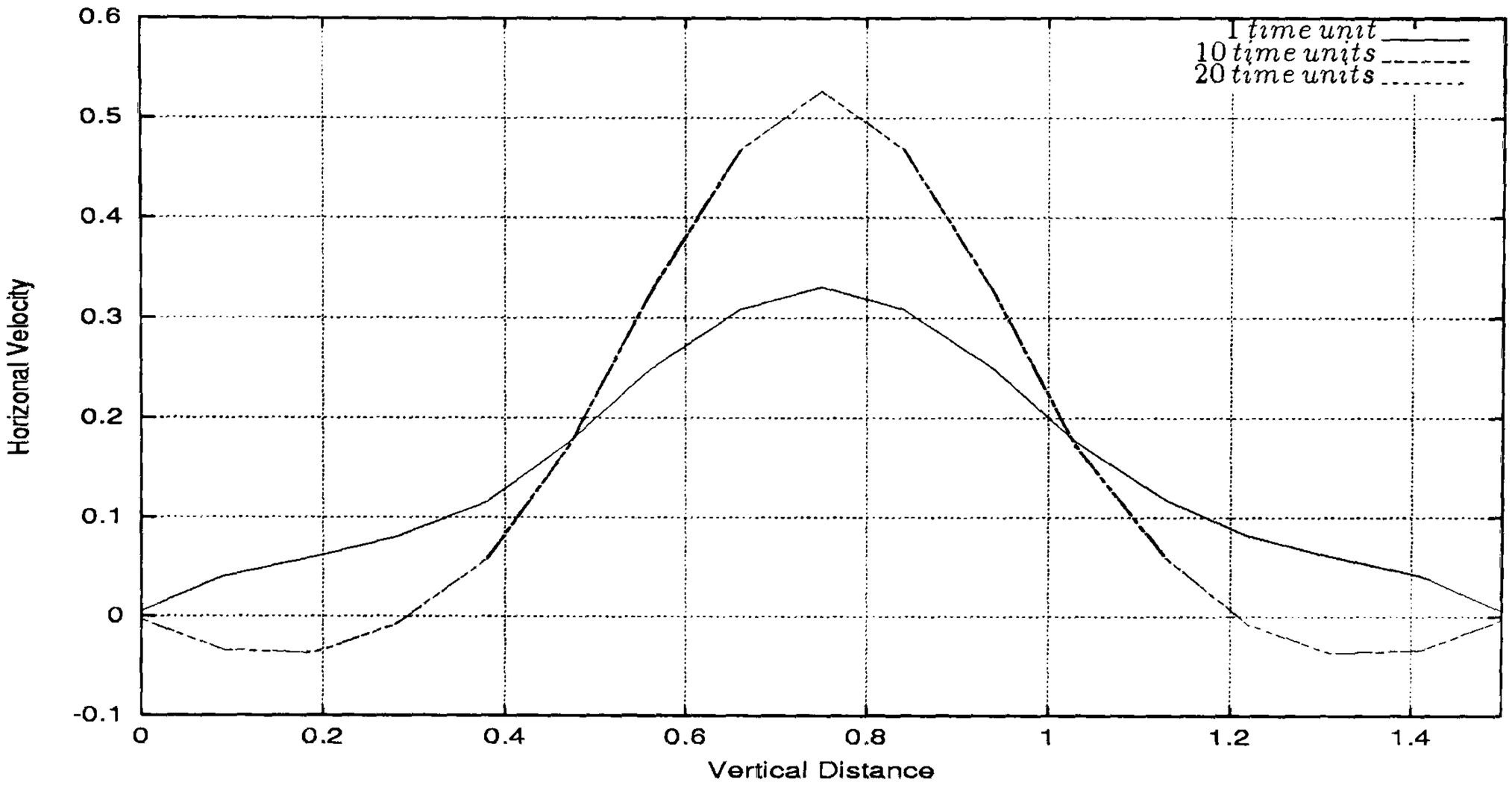


Figure 5.43: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 12.375$ for a 3 level computation. It can be seen the negative velocity values near the walls at 10 and 20 time units.

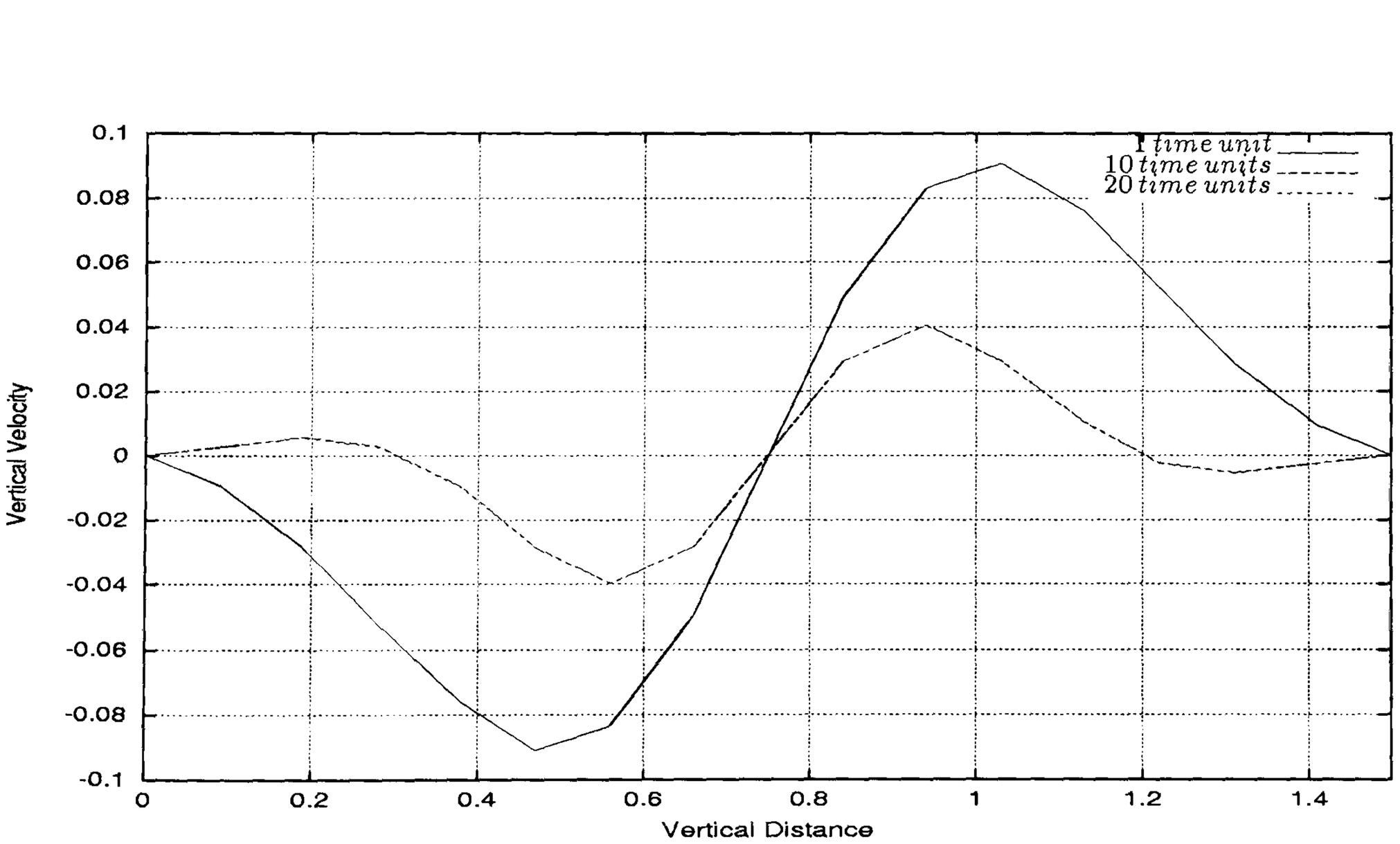


Figure 5.44: Single-phase flow through a contraction - vertical velocity profile along the line $x = 12.375$ for a 3 level computation. It can be observed the positive and negative values of the vertical velocity along this line.

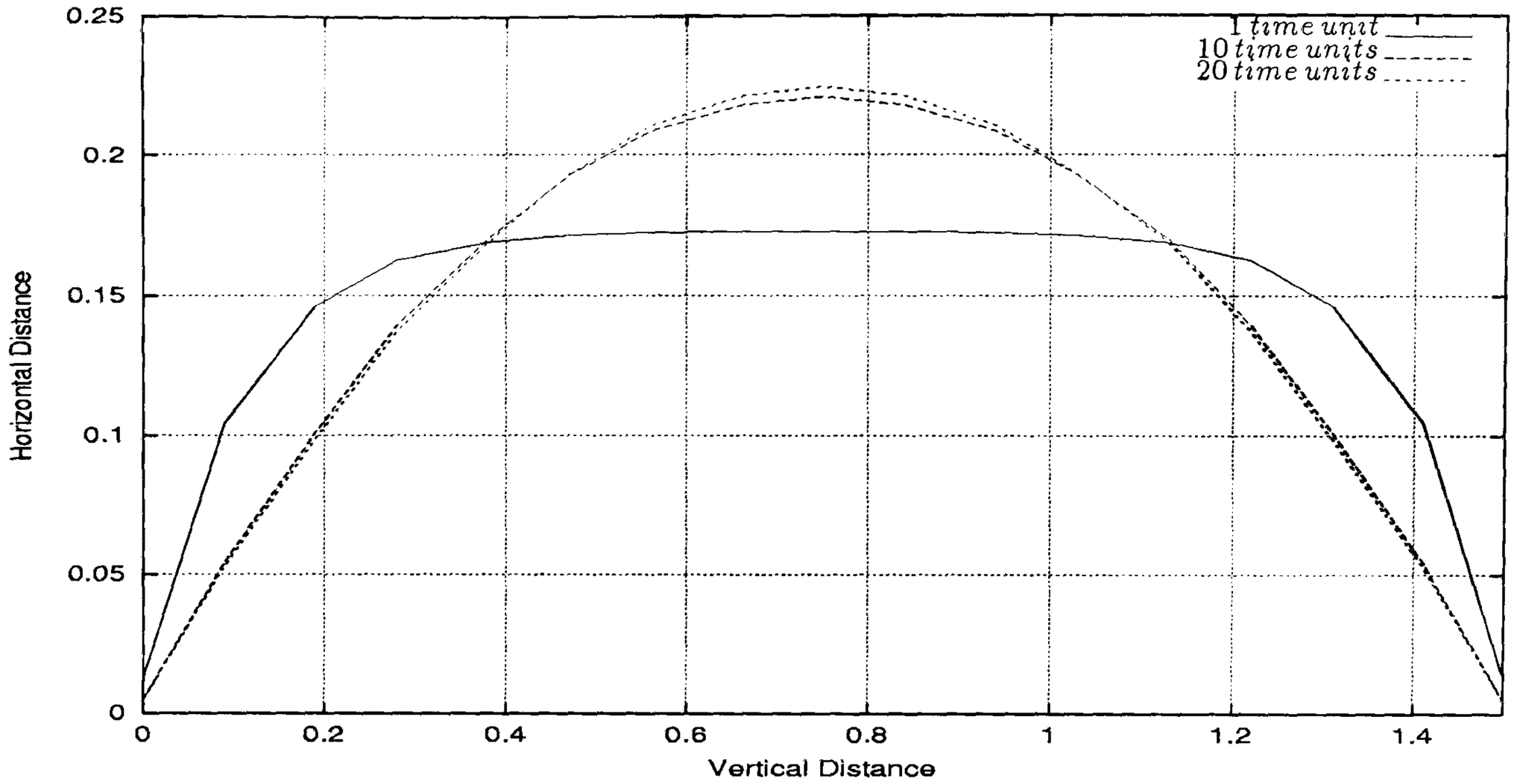


Figure 5.45: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 23.25$ for a 3 level computation. It can be seen the fully developed parabolic profile at 10 and 20 time units.

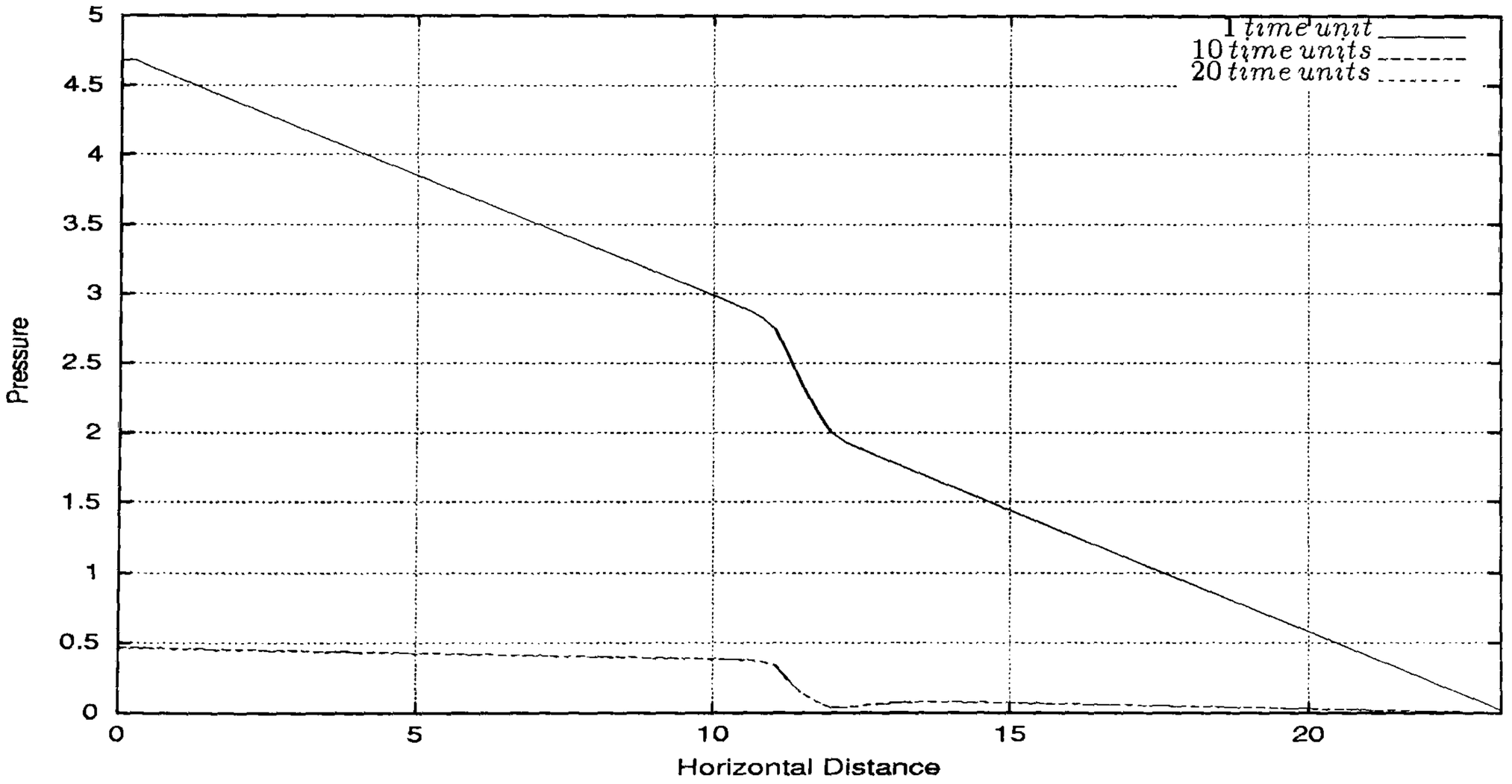


Figure 5.46: Single-phase flow through a contraction - pressure drop profile along the line $y = 0.75$ for a 3 level computation. It can be seen the increased pressure drop which occurs in the region of the contraction.

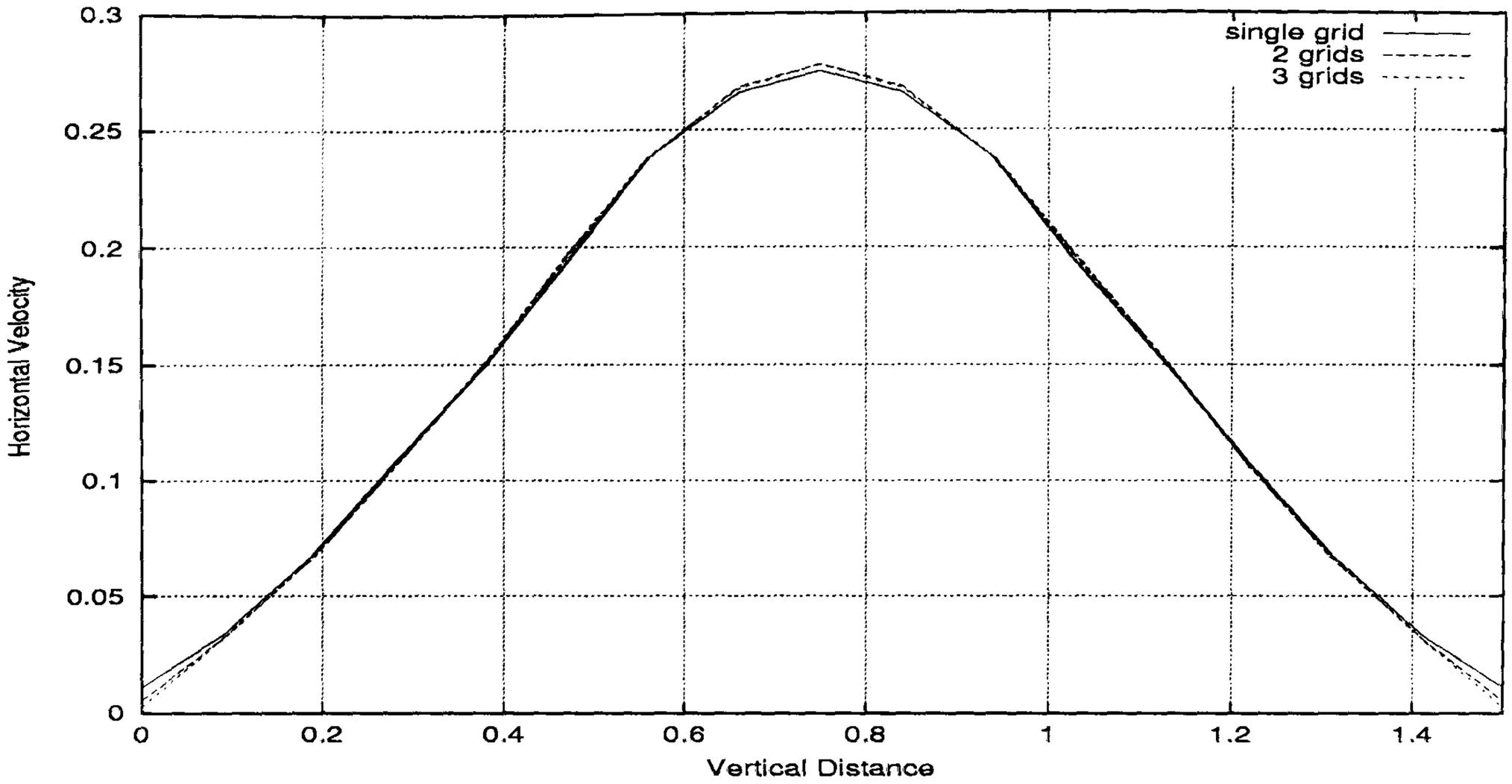


Figure 5.47: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 10.875$ for single 2 and 3 level computation at 20 time units. It can be seen good agreement between the parabolic profiles for all level calculations.

Grid and Time Step Independence of the Solutions

Figures 5.47 through 5.52 show for single 2 and 3 level computations the profiles for the horizontal velocity, vertical velocity and pressure drop along the selected sections of the domain at 20 time units. Since the profiles for the different grid computations show good agreement in all of the chosen sections it can be settled the grid independence of the solutions.

Figures 5.53 through 5.58 show for different time steps (1.0 and 10.0) the profiles for the horizontal velocity, vertical velocity and pressure drop along the selected sections of the domain. Good agreement can be observed between the 1.0 and 10.0 time steps profiles.

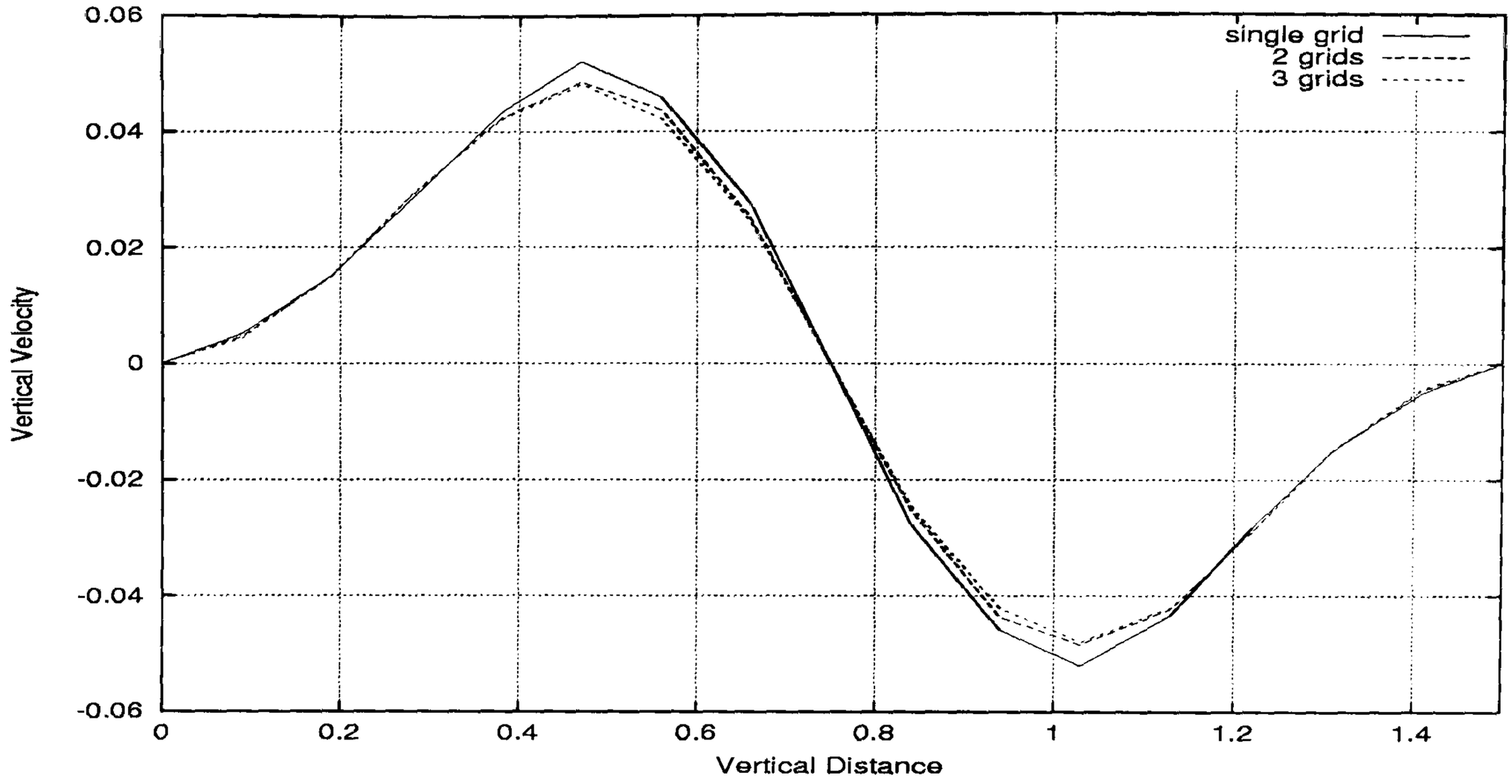


Figure 5.48: Single-phase flow through a contraction - vertical velocity profile along the line $x = 10.875$ for single 2 and 3 level computation at 20 time units. It can be observed the same pattern in the curves corresponding to all level calculations.

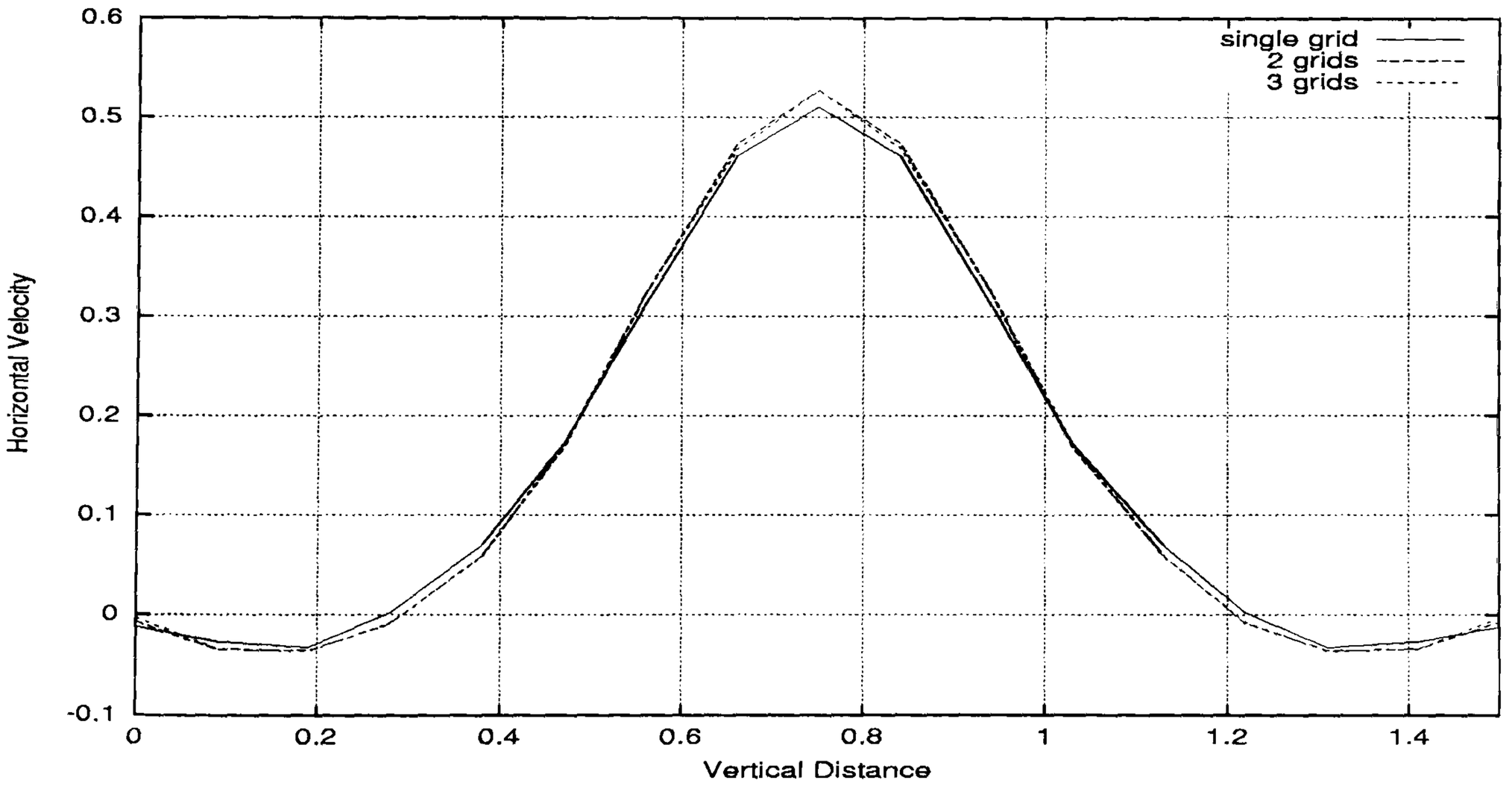


Figure 5.49: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 12.375$ for single 2 and 3 level computation at 20 time units. It can be seen good agreement between all the level computations.

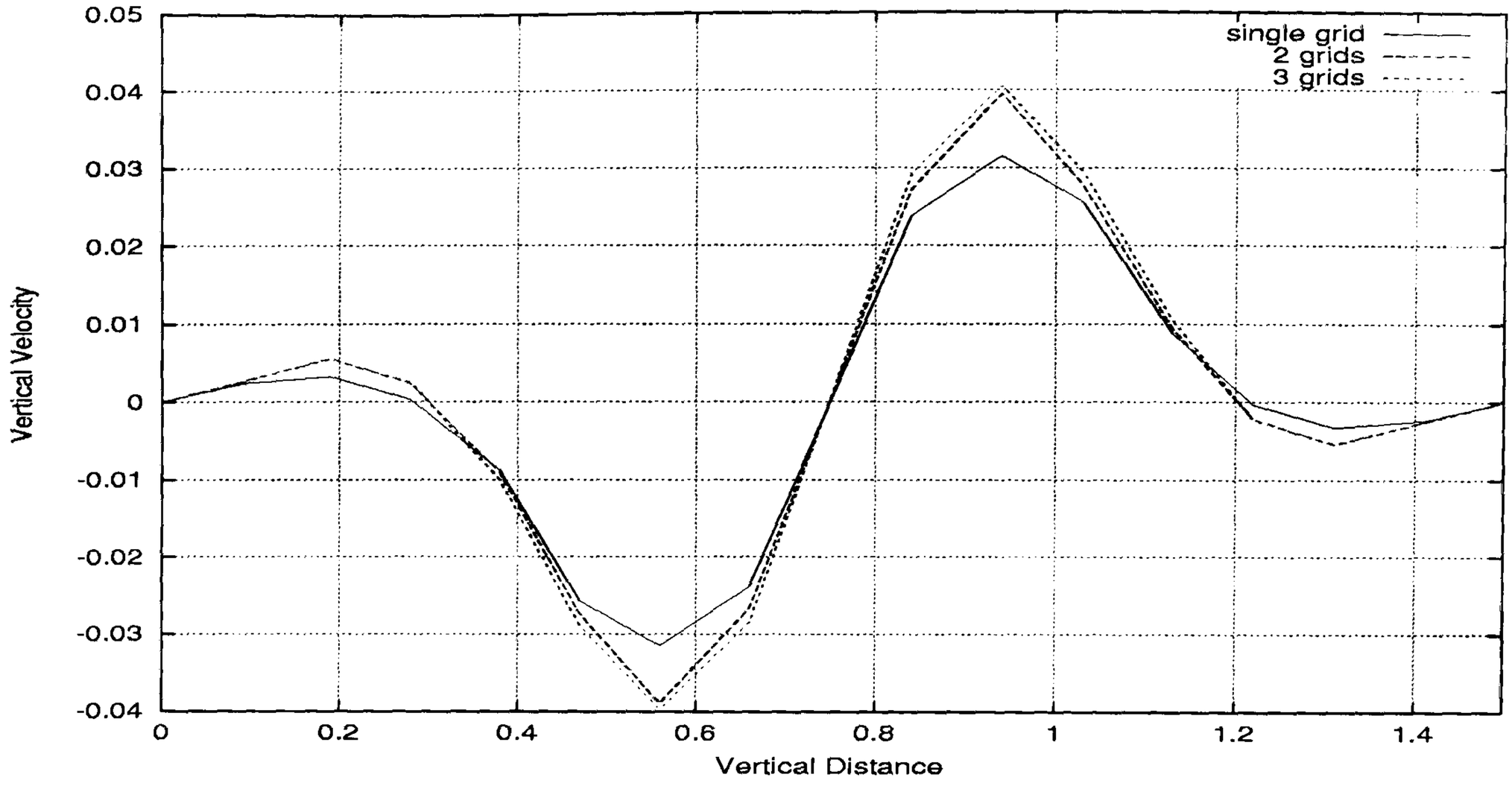


Figure 5.50: Single-phase flow through a contraction - vertical velocity profile along the line $x = 12.375$ for single 2 and 3 level computation at 20 time units. It can be observed good agreement between the profiles correspondent to 2 and 3 level computations.

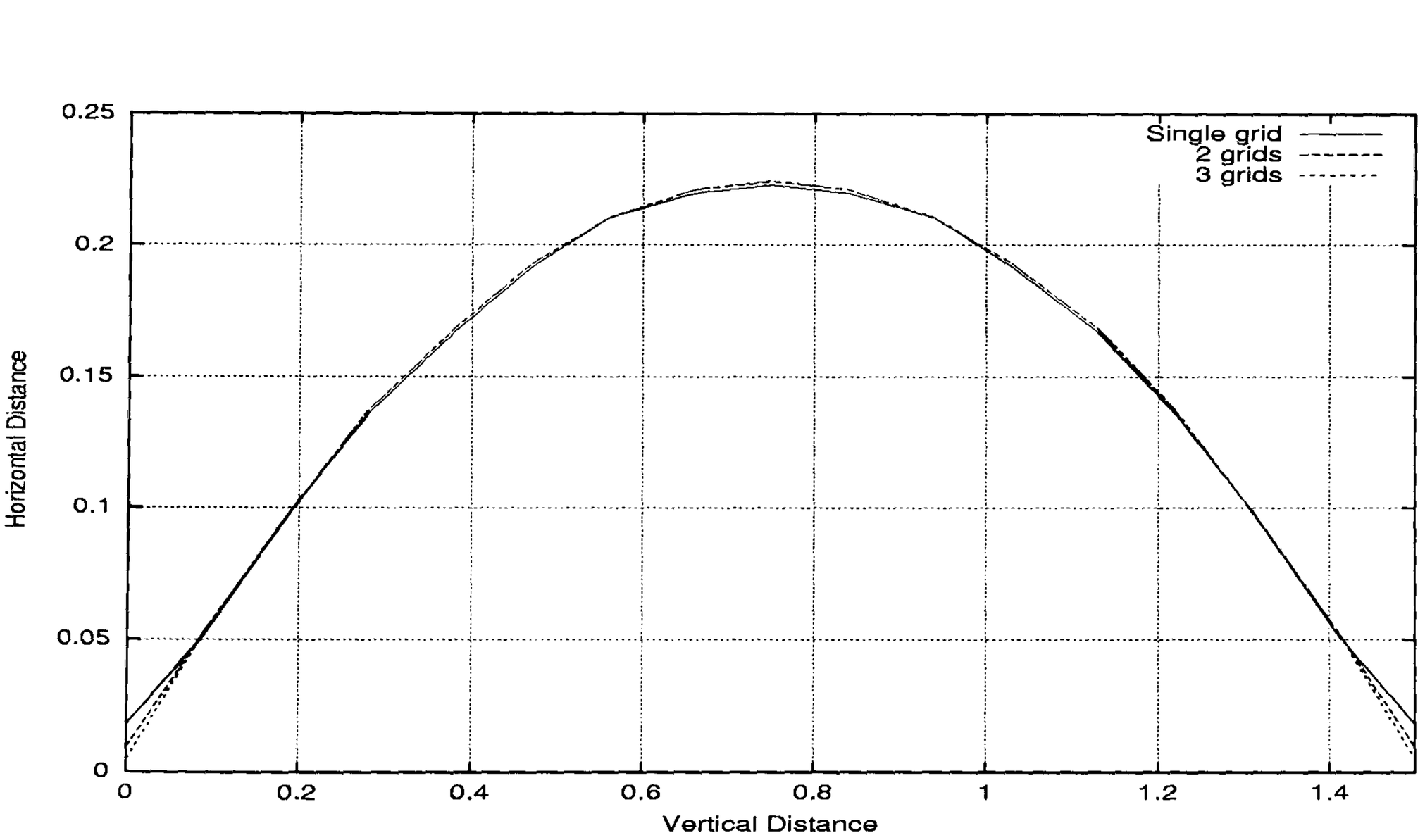


Figure 5.51: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 23.25$ for single 2 and 3 level computation at 20 time units. It can be seen good agreement between the fully developed parabolic profiles for all level computations.

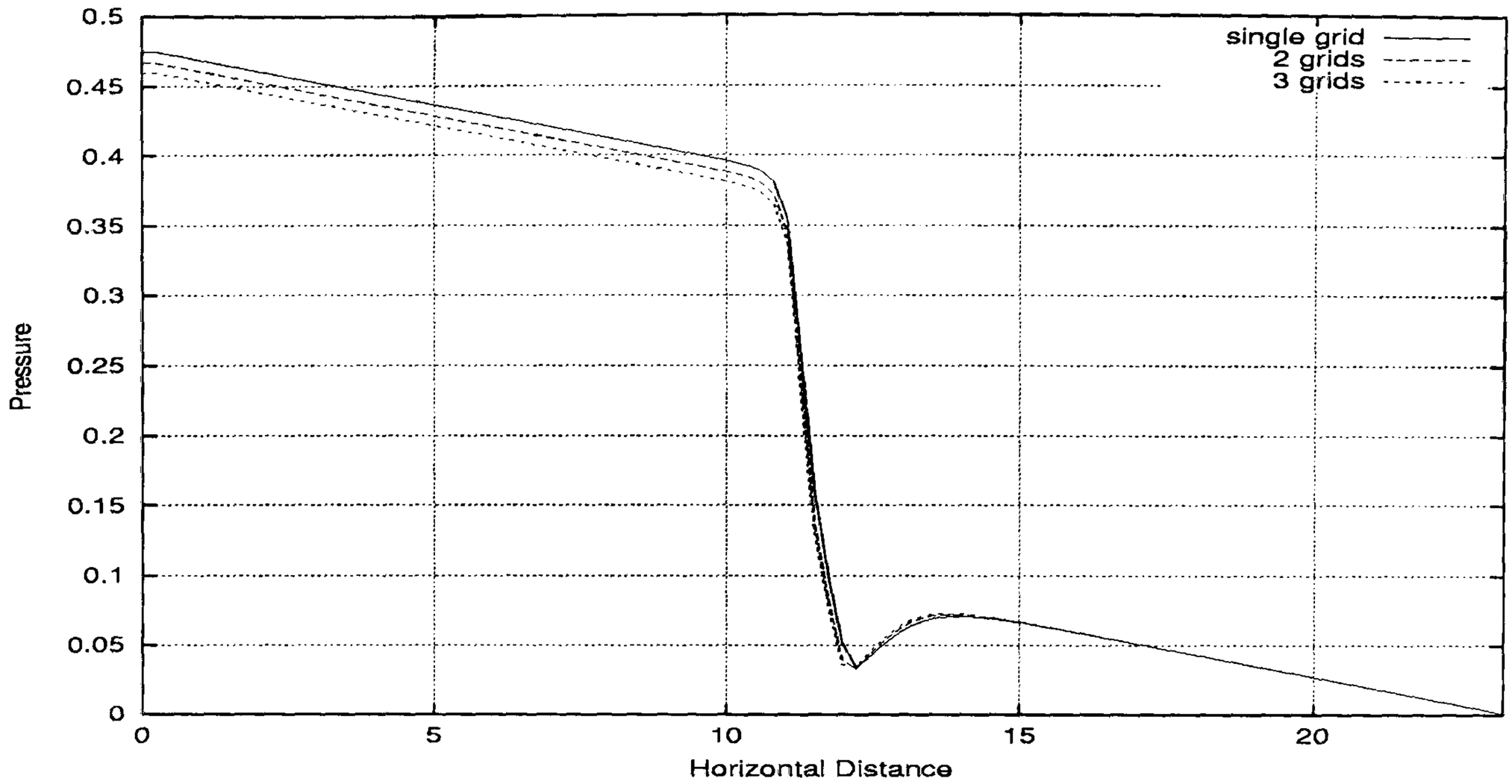


Figure 5.52: Single-phase flow through a contraction - pressure drop profile along the line $y = 0.5$ for single 2 and 3 level computation at 20 time units. It can be verified the similar pressure drop for all level calculations.

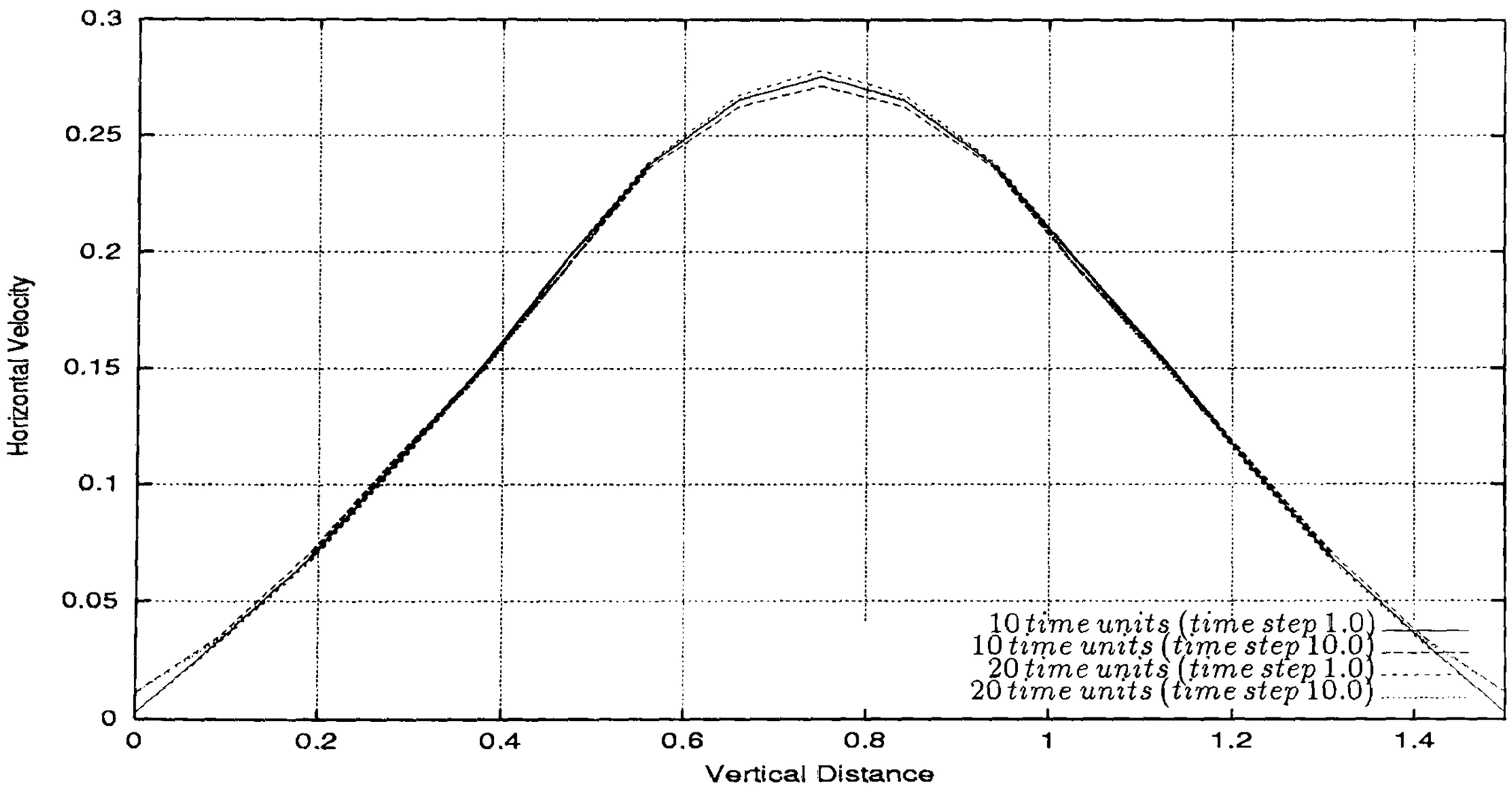


Figure 5.53: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 10.875$ for 3 level computation for different time steps. It can be seen good general agreement for all time steps computations.

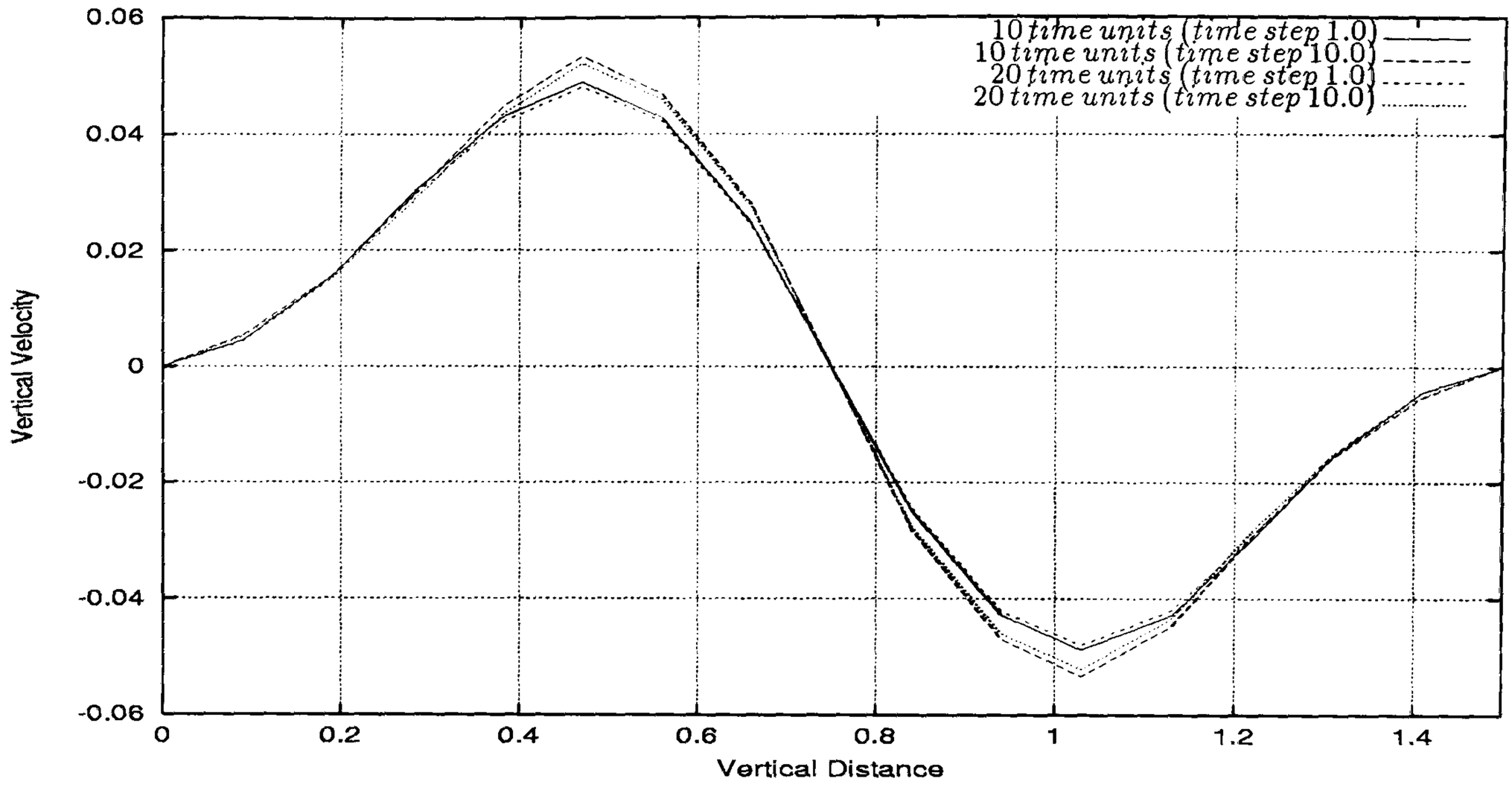


Figure 5.54: Single-phase flow through a contraction - vertical velocity profile along the line $x = 10.875$ for 3 level computation for different time steps. Good general agreement can be observed for all time steps calculations.

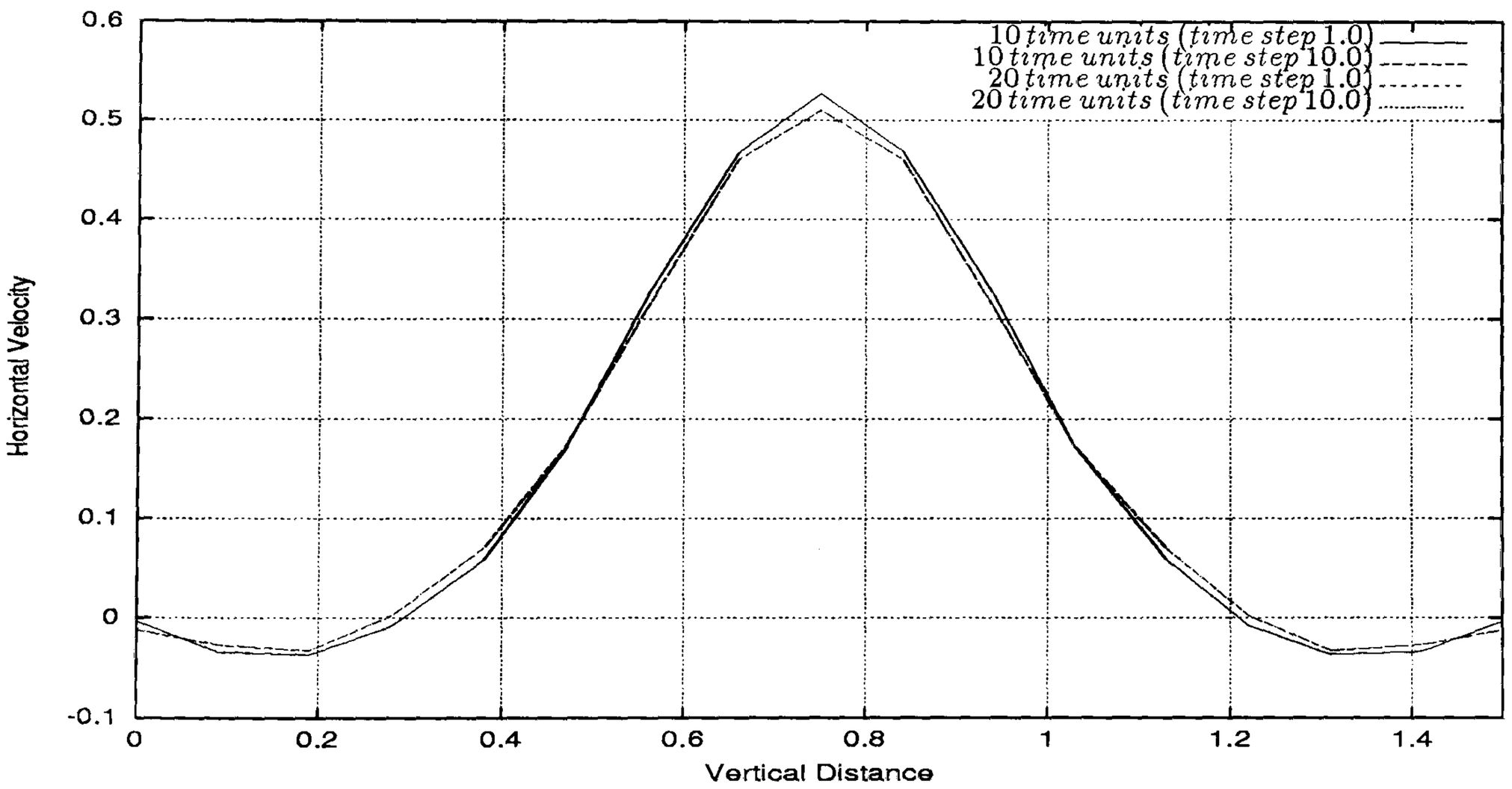


Figure 5.55: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 12.375$ for 3 level computation for different time steps. It can be observed good agreement for all time steps computations.

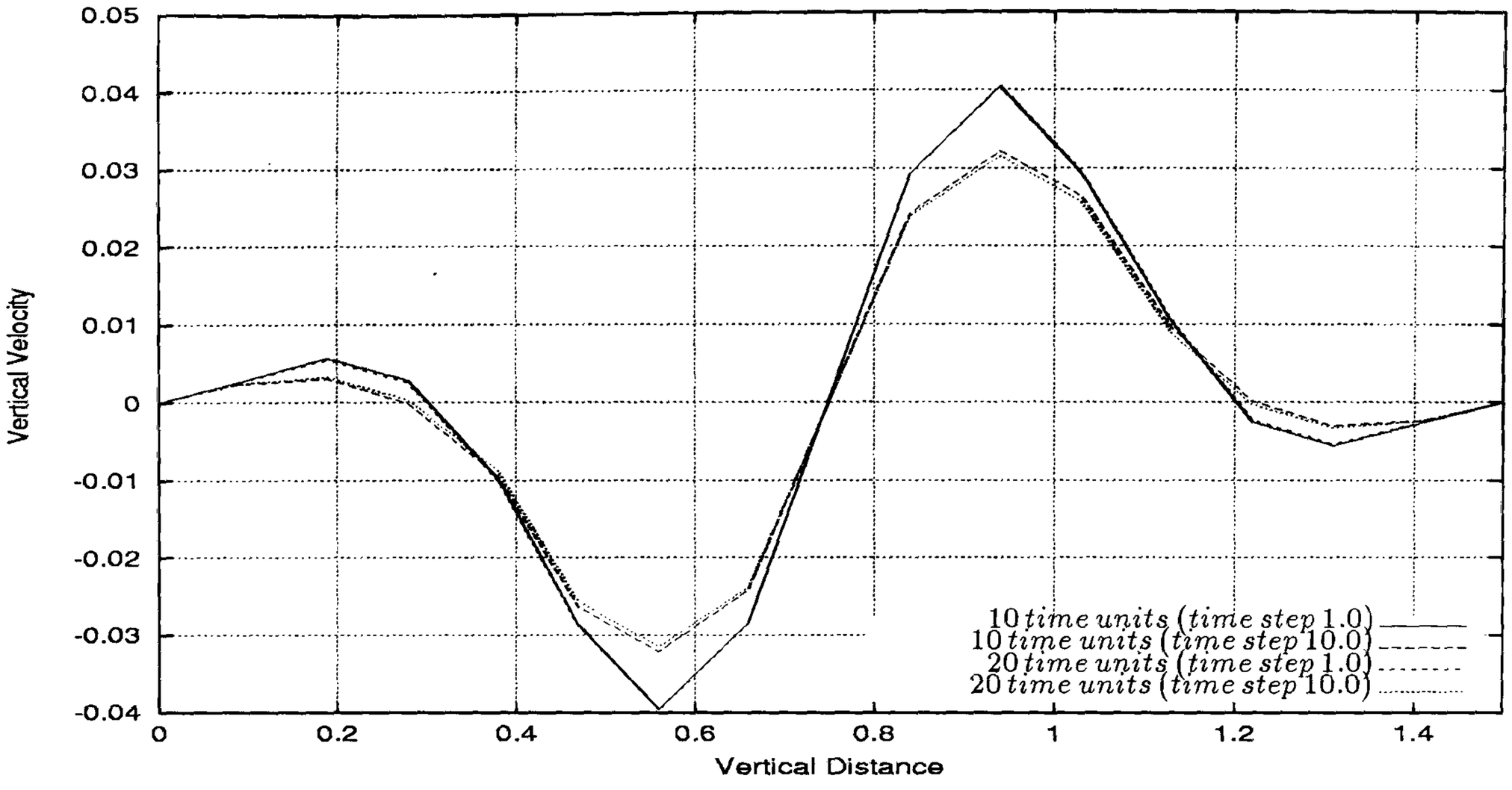


Figure 5.56: Single-phase flow through a contraction - vertical velocity profile along the line $x = 12.375$ for 3 level computation for different time steps. Good agreement can be observed for the vertical velocities at 20 time units.

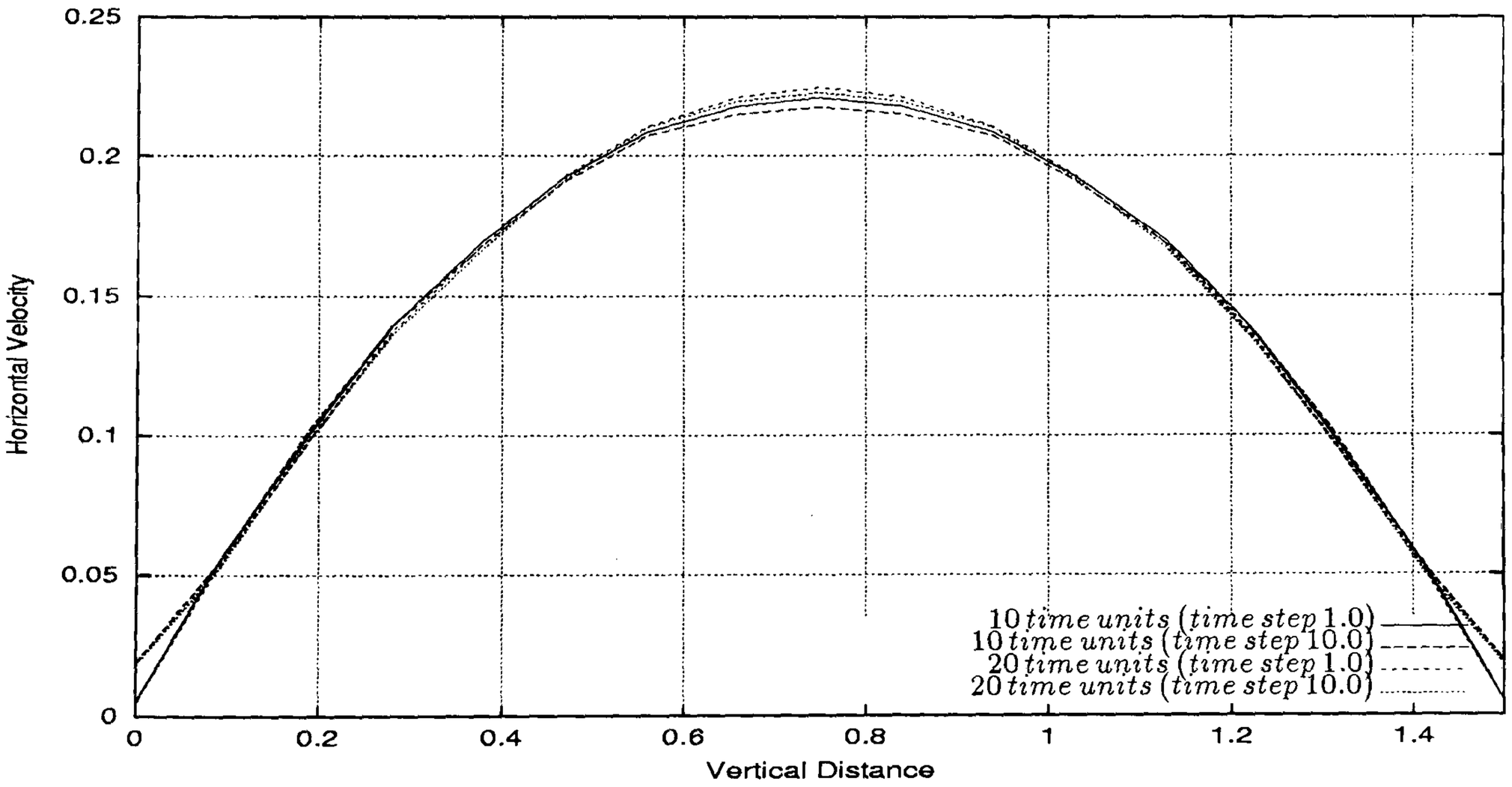


Figure 5.57: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 23.25$ for 3 level computation for different time steps. It can be observed good agreement between the fully developed parabolic velocity profiles at 10 and 20 time units.

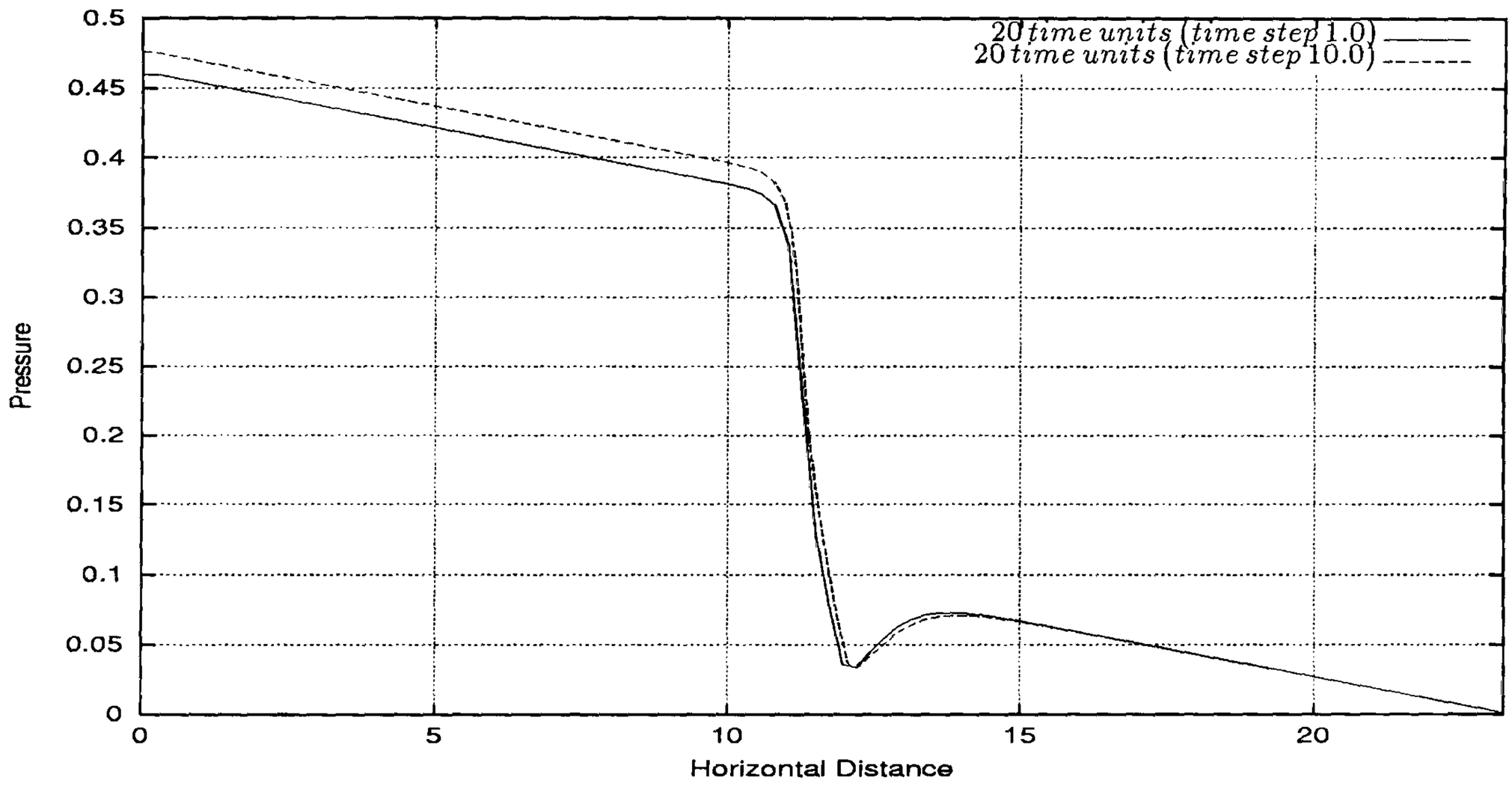


Figure 5.58: Single-phase flow through a contraction - pressure drop profile along the line $y = 0.75$ for 3 level computation for different time steps. It can be seen good agreement between the pressure drop computed by the two different ways.

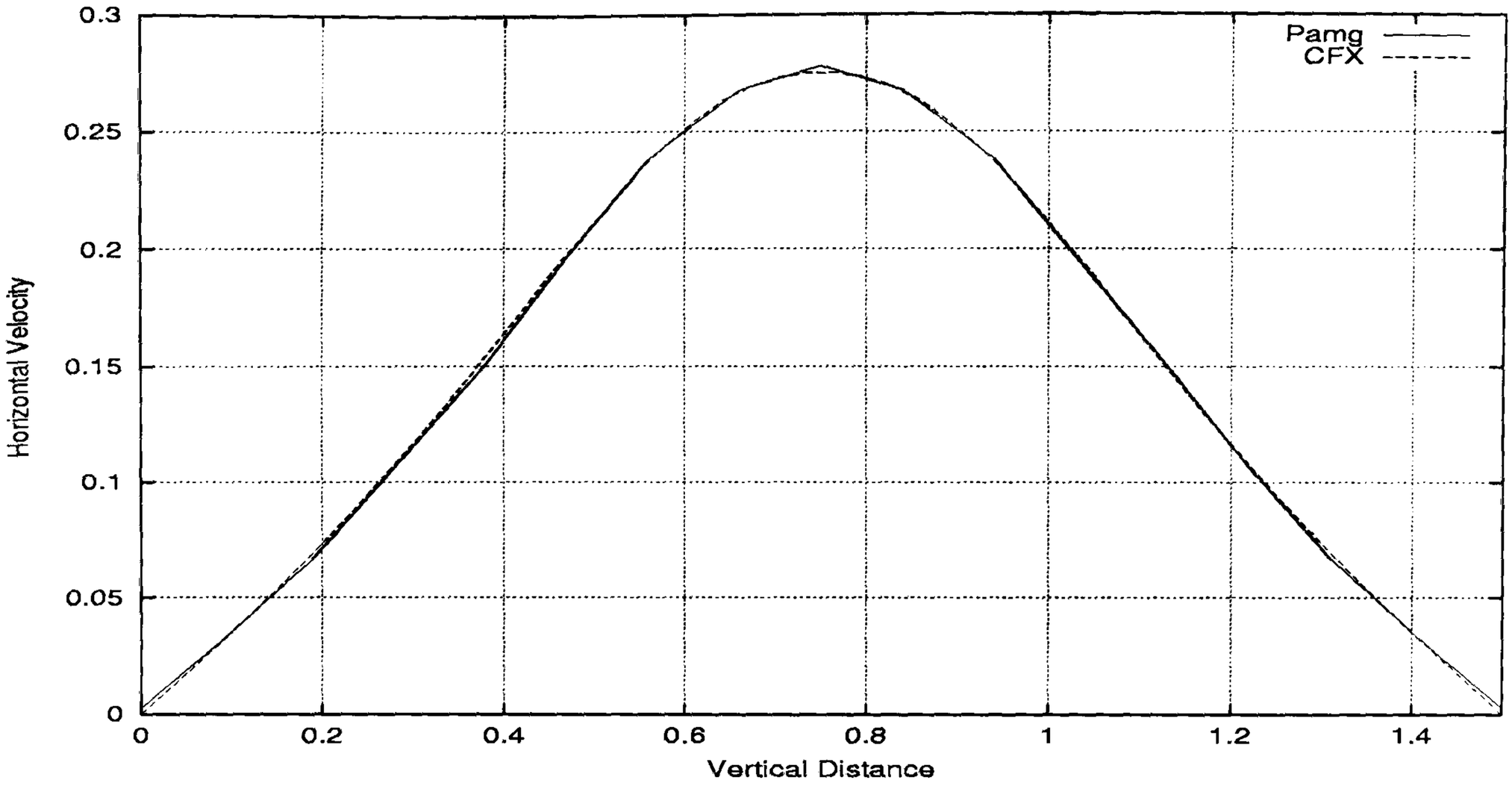


Figure 5.59: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 10.875$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be seen excellent agreement between the parabolic velocity profiles computed by both solvers.

Comparison with `CFX 4.3` Solutions

Figures 5.59 through 5.65 show the comparison between `transient Pamg Single-phase` and the `CFX 4.3` solver for this particular test case. The agreement between the solution profiles is almost perfect. Figure 5.65 which represents the comparison between the pressure drop along the line $y = 0.75$ at 20 time units strongly suggests this.

Summary

To finish the validation process of the `transient Pamg Single-phase` we outlined this last test case of a single-phase flow through a contraction. This is the more complex domain that we have considered and consequently the validation of the solver in this geometry completes in a very effective way this process. Has already have been observed in the expansion test case and here becomes more evident the solver depends on the Reynolds number of the fluid flow. Consequently care had to be undertaken in the choice of the Reynolds number of the flow.

The grid independence of the solutions have been verified with experiments in single 2 and 3 level computations. The profiles plotted in the chosen lines show this happen. The independence of the solver on the time step has been also addressed in experiments with different time steps. Consequently it can be considered that the `transient Pamg Single-phase` does not depend on the time step. This is precisely what is expected for flows that are expected to reach the steady state regime. Time step independence for genuinely unsteady flows has not been tested since it was not in the scope of this thesis.

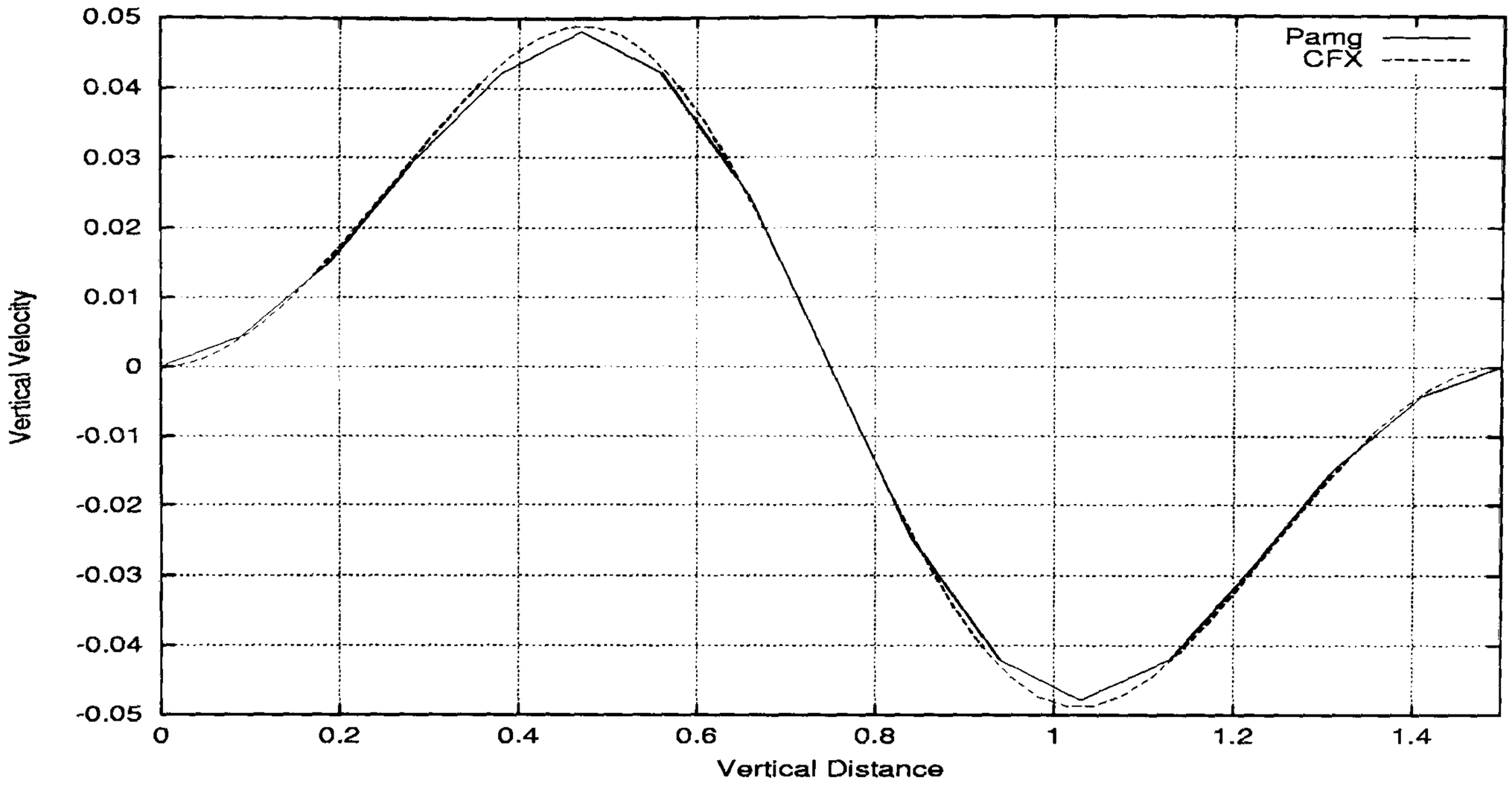


Figure 5.60: Single-phase flow through a contraction - vertical velocity profile along the line $x = 10.875$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be observed good agreement between the vertical velocity profiles computed by both solvers.

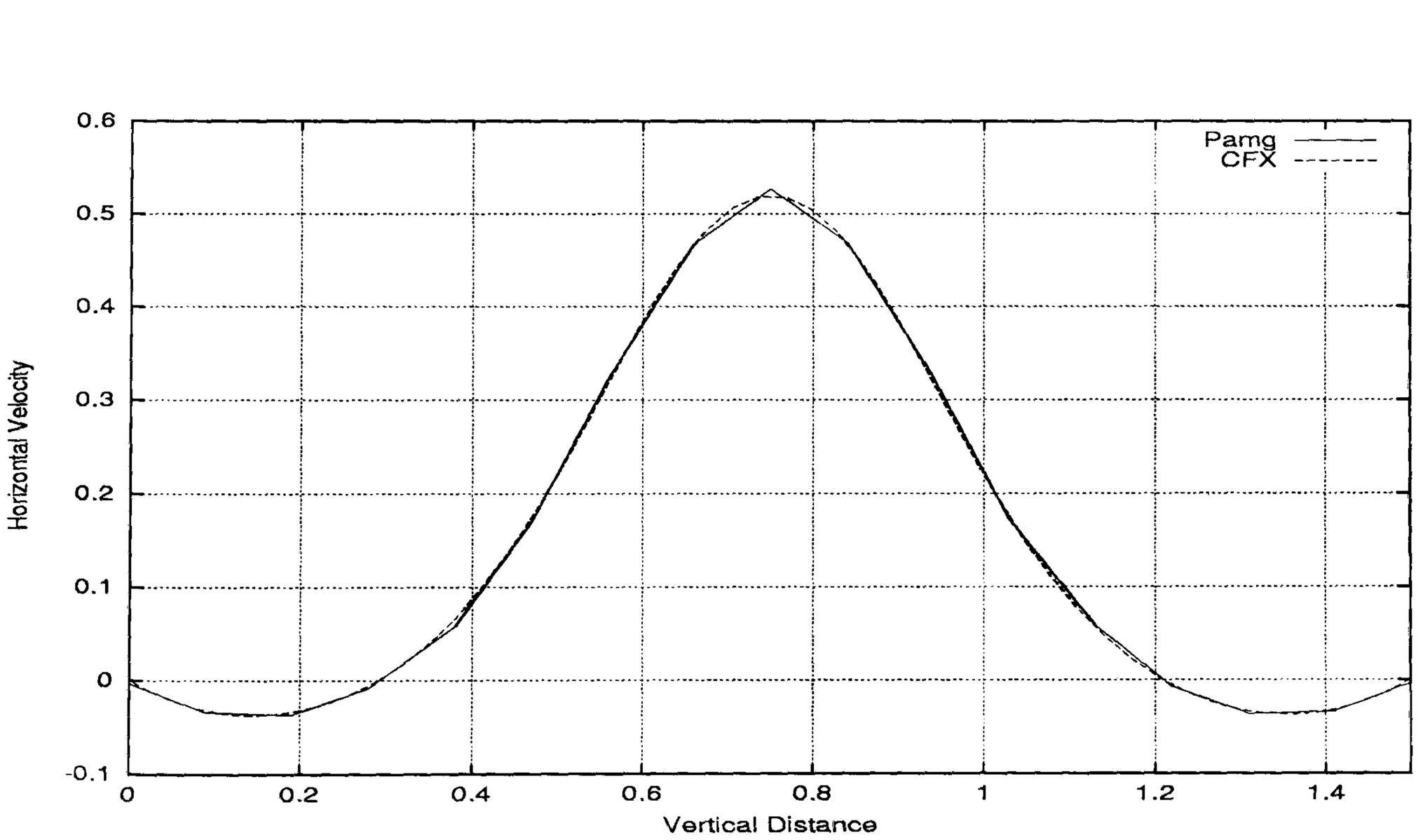


Figure 5.61: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 12.375$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. Good agreement can be observed in the profiles produced by both solvers.

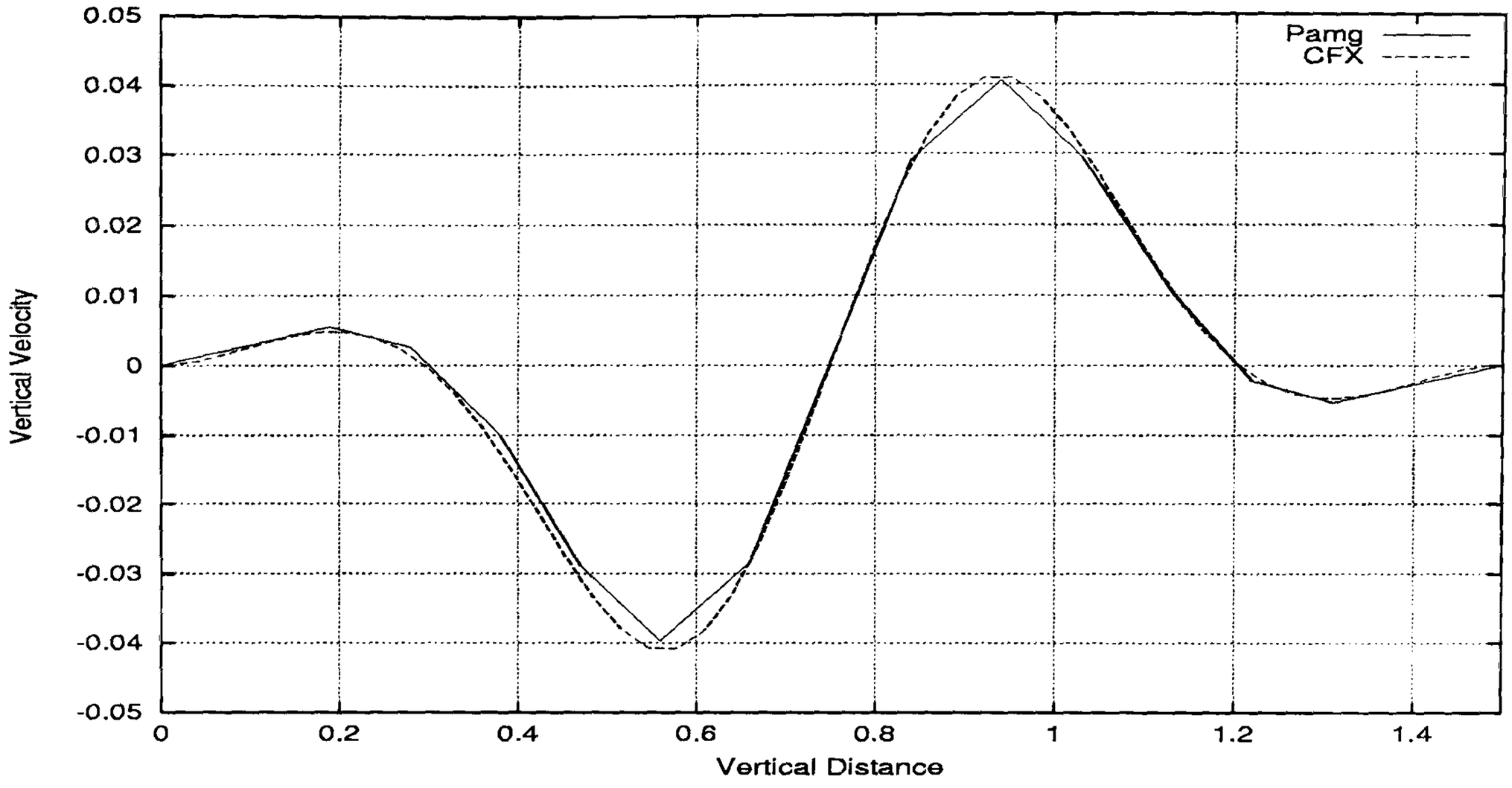


Figure 5.62: Single-phase flow through a contraction - vertical velocity profile along the line $x = 12.375$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be observed general good agreement between the vertical velocities computed by both solvers.

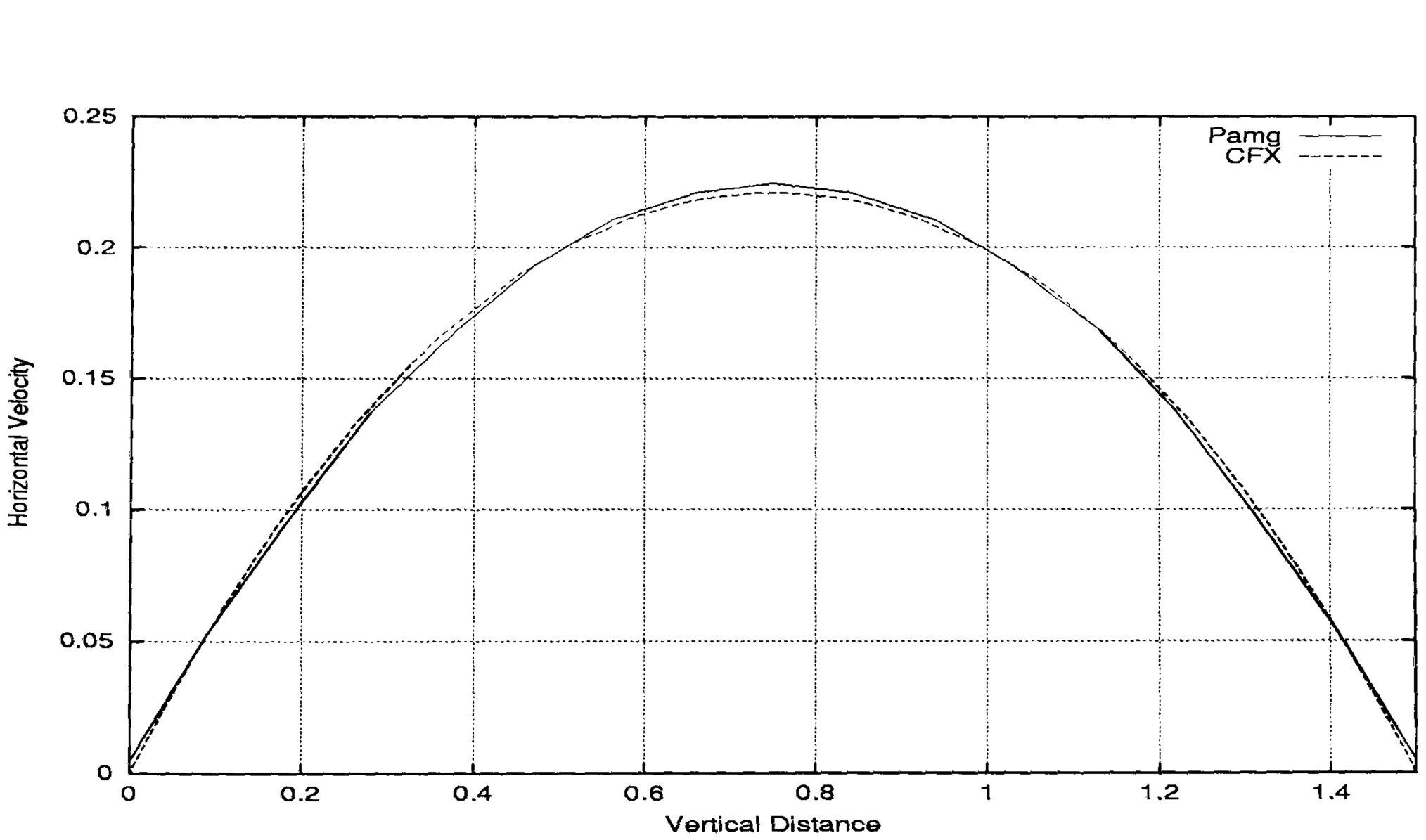


Figure 5.63: Single-phase flow through a contraction - horizontal velocity profile along the line $x = 23.25$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be seen good agreement in the fully developed parabolic velocity profiles produced by both solvers.

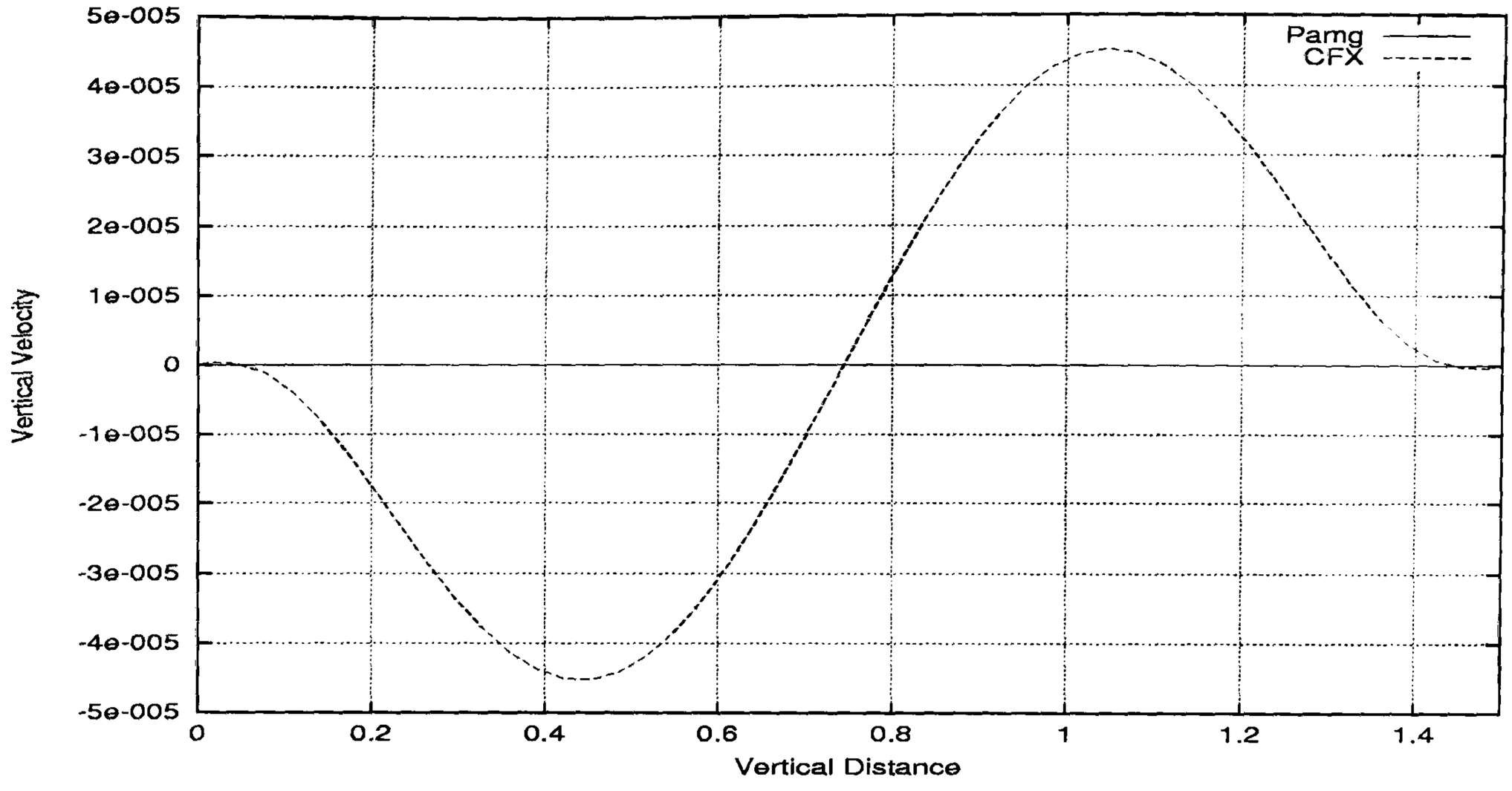


Figure 5.64: Single-phase flow through a contraction - vertical velocity profile along the line $x = 23.25$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be seen that the vertical velocities produced by both solvers are close to zero. Note the scale of the vertical velocity axis.

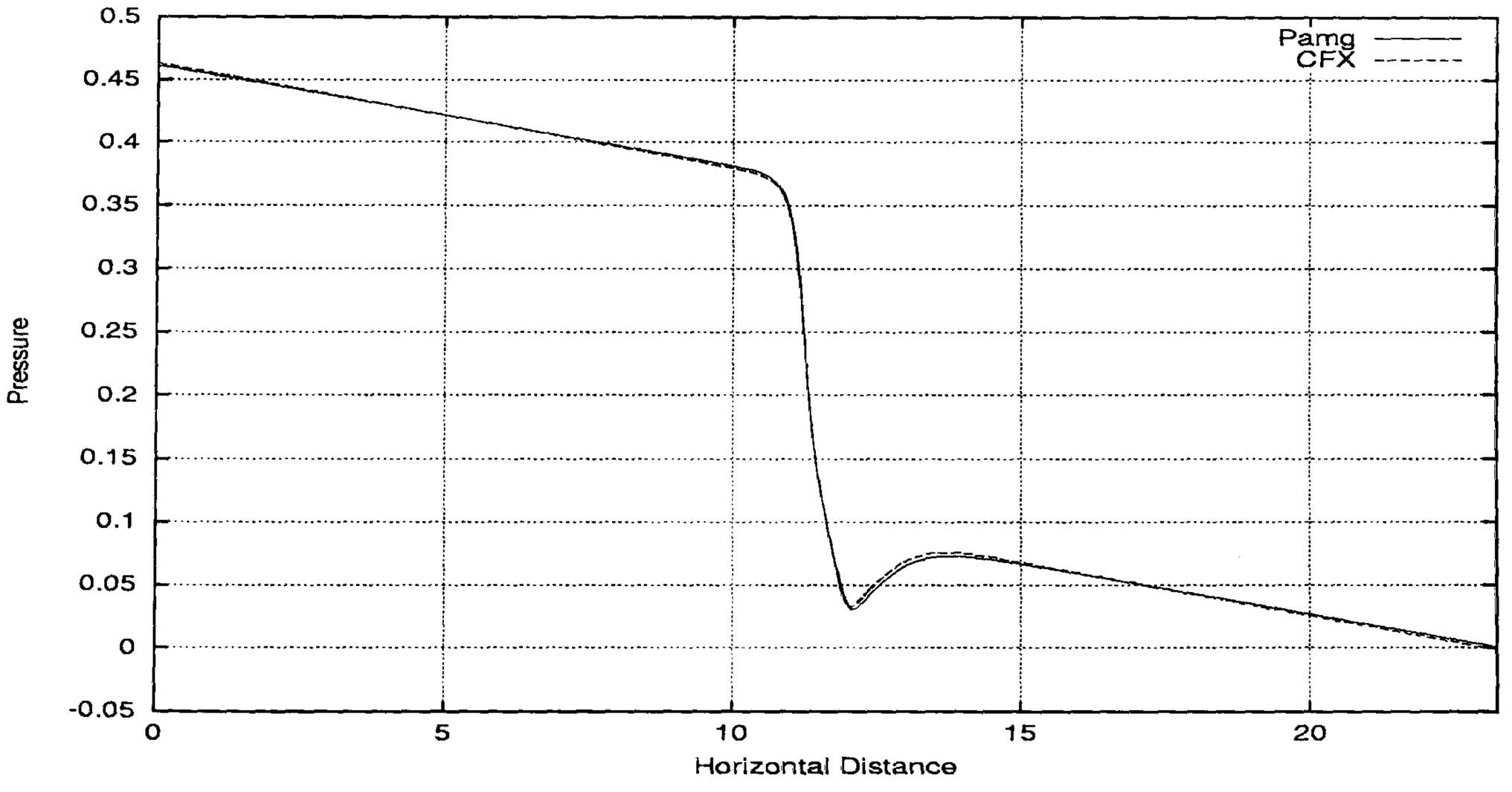


Figure 5.65: Single-phase flow through a contraction - pressure drop comparison along the line $y = 0.75$ for `CFX 4.3` and `transient Pamg Single-phase` at 20 time units. It can be verified the good agreement between the pressure drop computed by both solvers.

Finally comparison of the results produced by `transient Pamg Single-phase` with the `CFX 4.3` solver have been outlined. Evidence suggests that the agreement between the two solvers is very positive even in this difficult problem and consequently the implementation of the `transient Pamg Single-phase` is correct.

5.3 Transient Pamg Multi-Phase. Validation Problems

5.3.1 Two-Phase Channel Flow

Problem definition

The first test problem to validate the `transient Pamg Multi-phase` is the simulation of two fluids in a channel. The geometry is the same has that one defined for the first test case of the `transient Pamg Single-phase` defined in Section 5.2.1. This test case has been chosen because is very simple and consequently a good choice as the first test problem to validate the transient multi-phase solver. The physical properties of the two fluids are

Two-Phase Flow	Reynolds Number	Viscosity	Density
Phase I	100	0.01	1.0
Phase II	100	0.005	0.5

The values chosen for the drag coefficient and the inter-facial length (see Equation (3.43)) are:

$$C_D = 1.0$$

$$d_{\alpha\beta} = d_{\beta\alpha} = 0.1$$

The computational grids are the same as for the single-phase channel test case (see Figure 5.2). Naturally the sections where the solution profiles are going to be presented are exactly the same as in the single-phase case (see Figure 5.3).

The boundary conditions used are at the inlet, Dirichlet boundary conditions and at the outlet Neumann conditions. In the inlet a parabolic velocity profile is specified for the x-component velocity $u(x, y)$ of each phase by

$$u(0, y) = 4y(1 - y)$$

and the y-component velocity $v(x, y)$ of each phase is null ($v(0, y) = 0$). At the outlet the y-component velocity is null either.

The volume fractions in the inlet are set to 0.5 for each phase.

The initial conditions for the simulation are very simple. The velocities are all set to zero and the volume fractions to 0.5. The convergence tolerance for the simulation is 10^{-6} for each time step. Since `transient Pamg Multi-phase` is a non-dimensional solver we use like in the `transient Pamg Single-phase` solver the concept of time unit to specify the simulation at a specific time of integration.

Results for a 3 level computation

In Figures 5.66 and 5.67 we see the streamlines for the two phases after 20 time units. Quite naturally, these are straight horizontal lines. This is due to the fact

that the vertical velocity components are set to zero in the inlet and the walls are horizontal and parallel each other.

Figures 5.68 through 5.72 show the horizontal velocity, vertical velocity and volume fraction profiles for each phase along the lines $x = 2.625$ and $y = 0.5$ at 1 and 20 time units. It can be observed the fully developed parabolic profile of the horizontal velocities at 20 time units. The vertical velocites are very close to zero at 1 and 20 time units. The volume fractions start to be very different at 1 time unit and end approaching the value 0.5 at 20 time units.

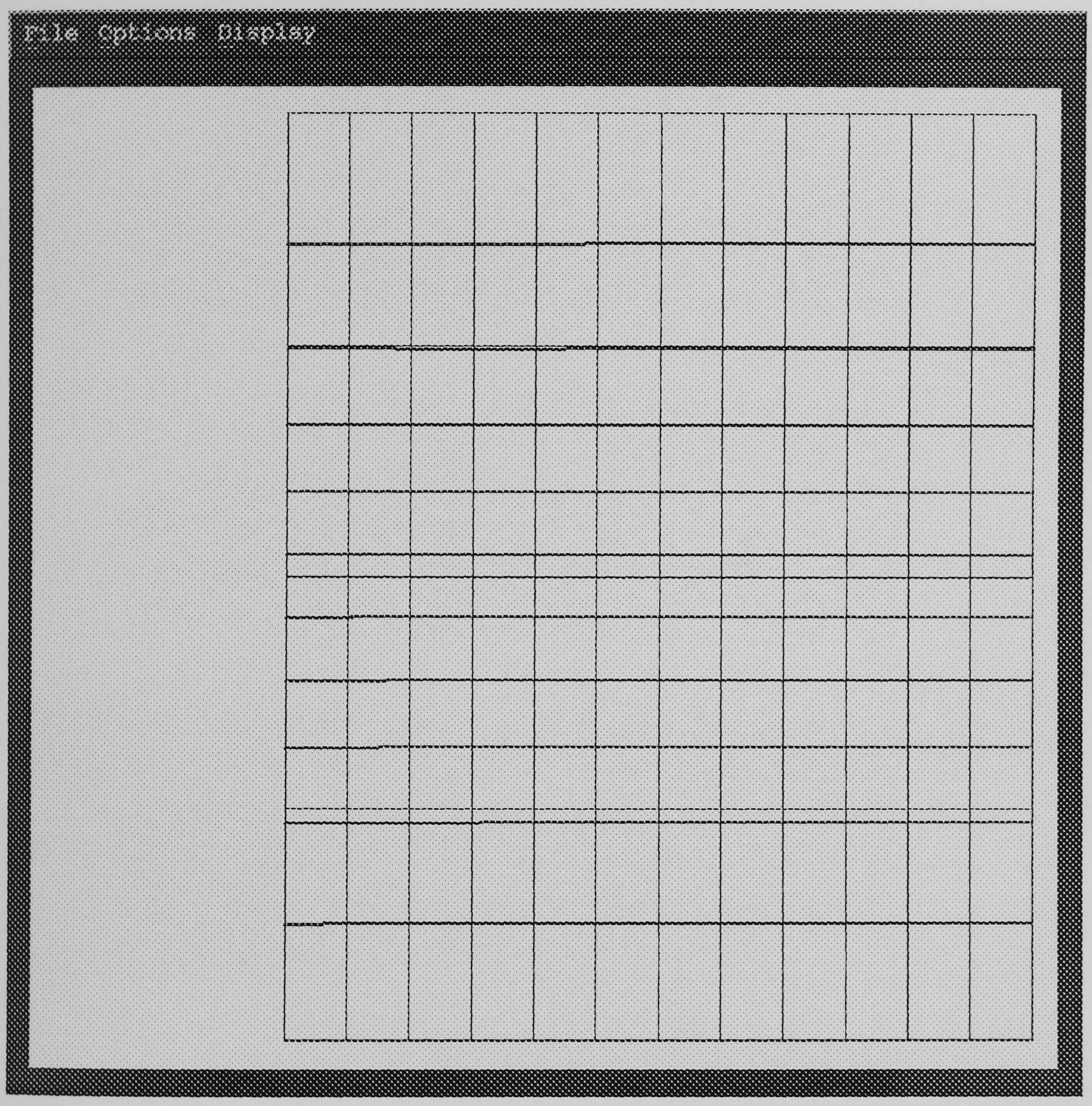


Figure 5.66: Multi-phase channel flow - streamlines after 20 time units for *Phase I*

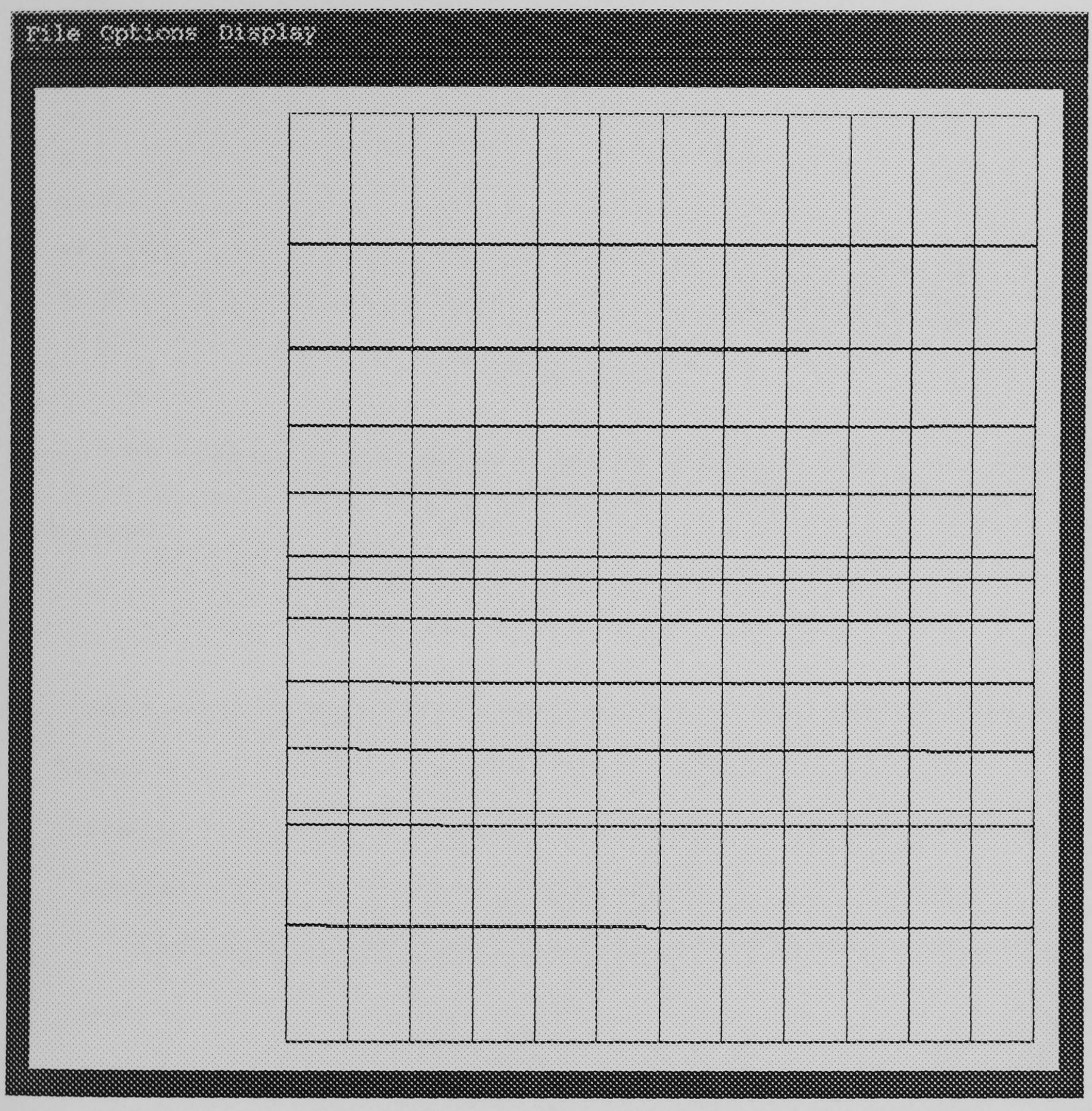


Figure 5.67: Multi-phase channel flow - streamlines after 20 time units for *Phase II*

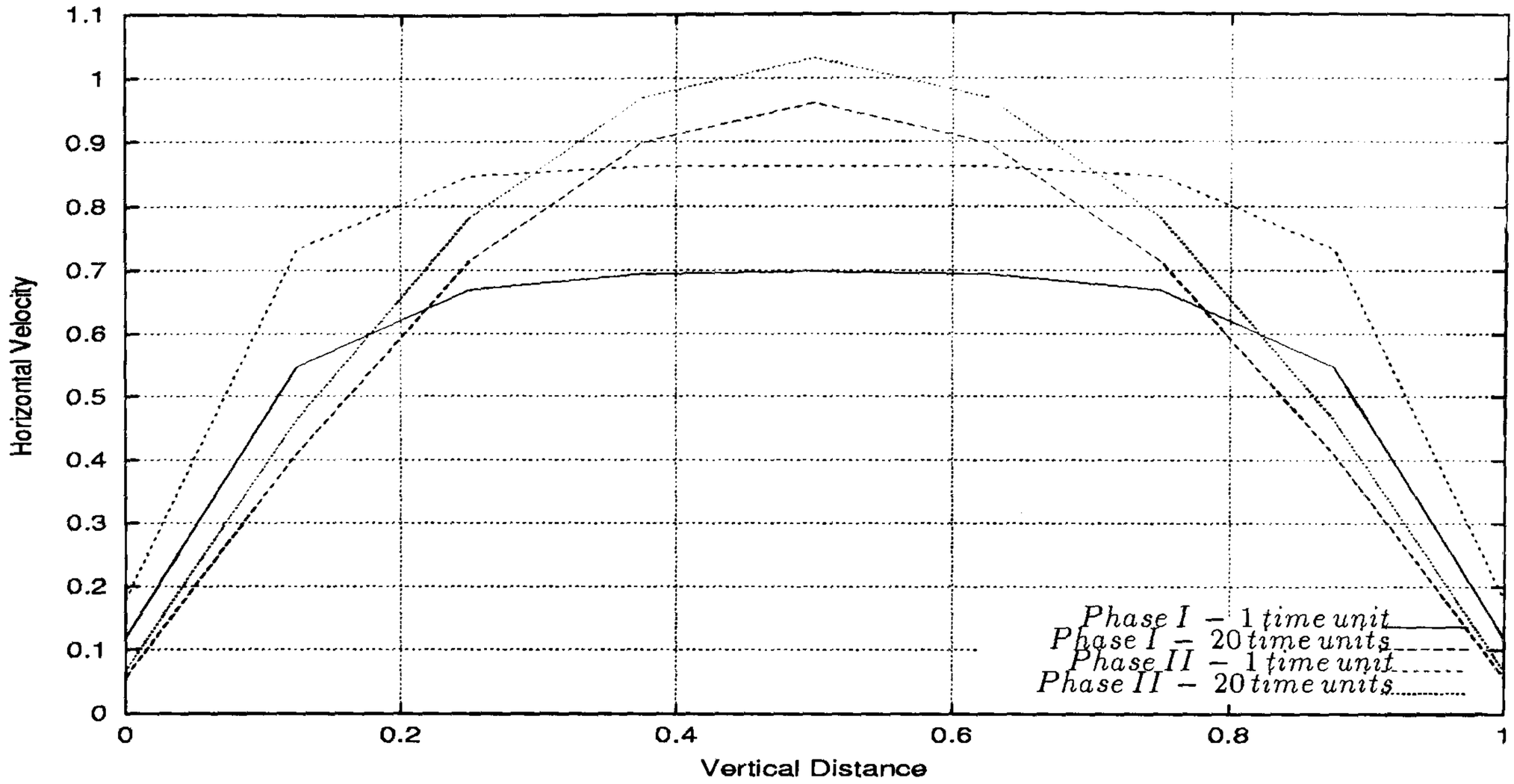


Figure 5.68: Multi-phase channel flow - horizontal velocity profile along the line $x = 2.625$. It can be seen the fully developed parabolic profile at 20 time units for both phases.

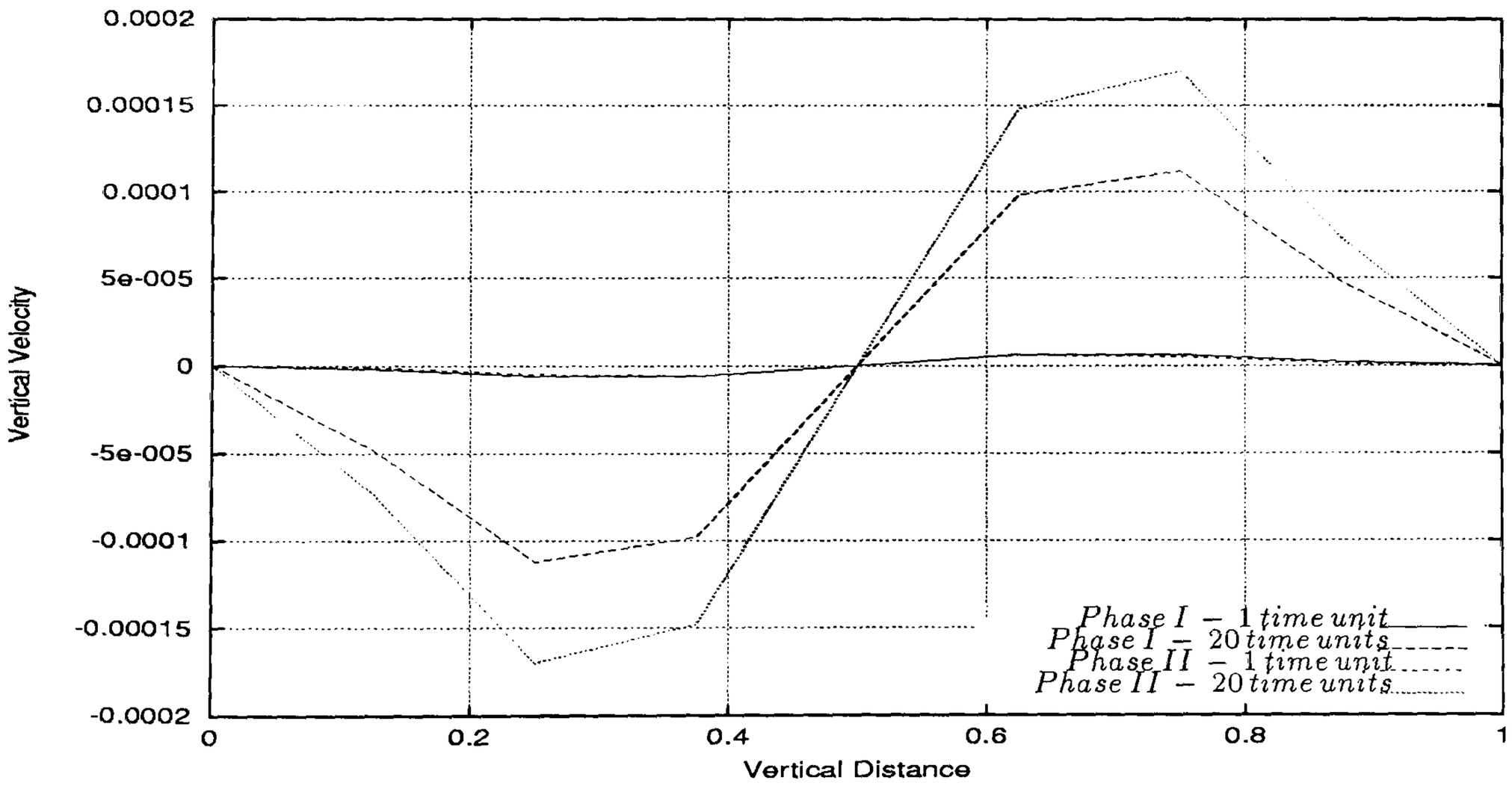


Figure 5.69: Multi-phase channel flow - vertical velocity profile along the line $x = 2.625$. It can be observed that the vertical velocities are very close to null for both phases. Note the small scale of the vertical velocity axis.

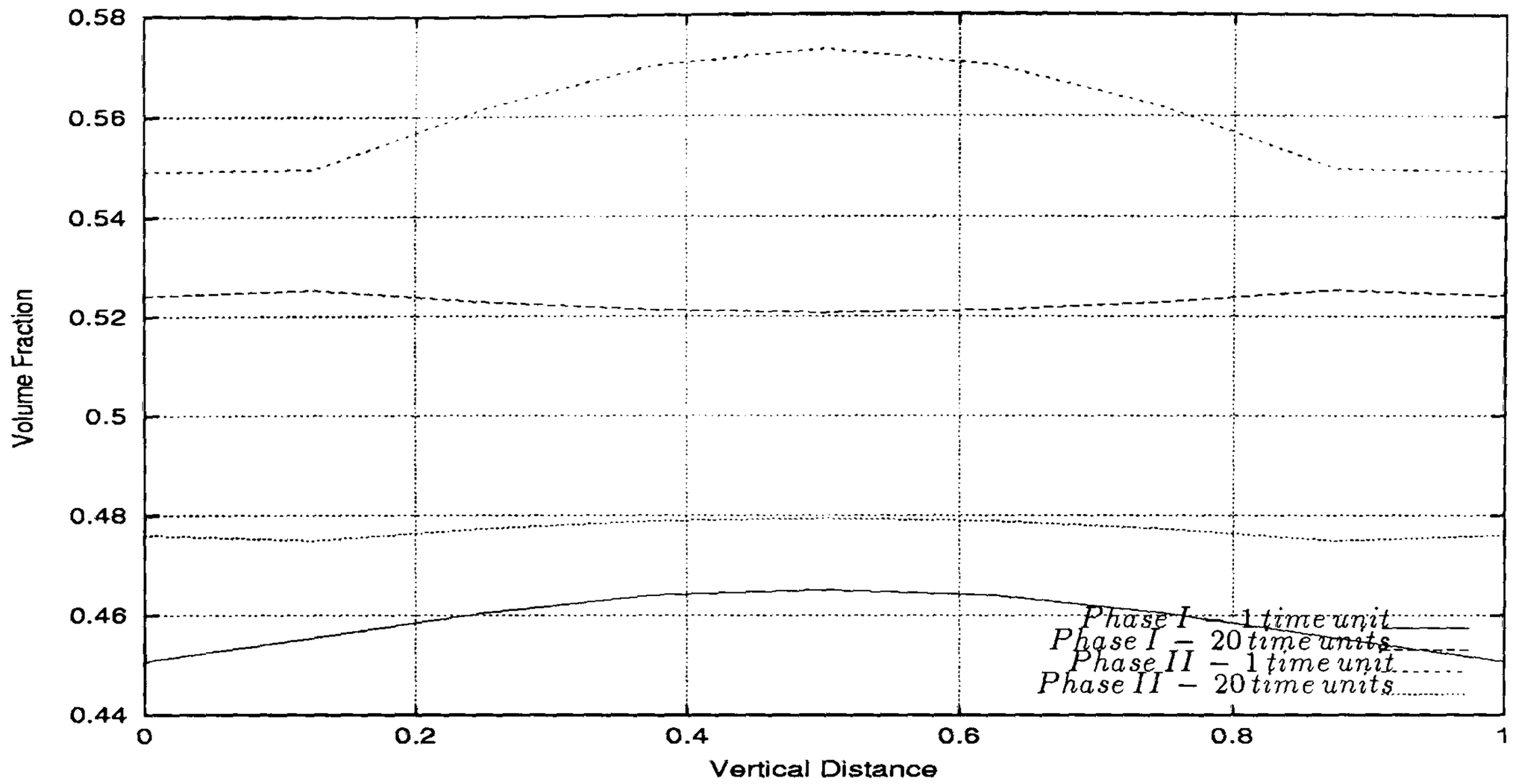


Figure 5.70: Multi-phase channel flow - volume fraction profile along the line $x = 2.625$. It can be observed the variation of the volume fractions from 1 to 20 time units.

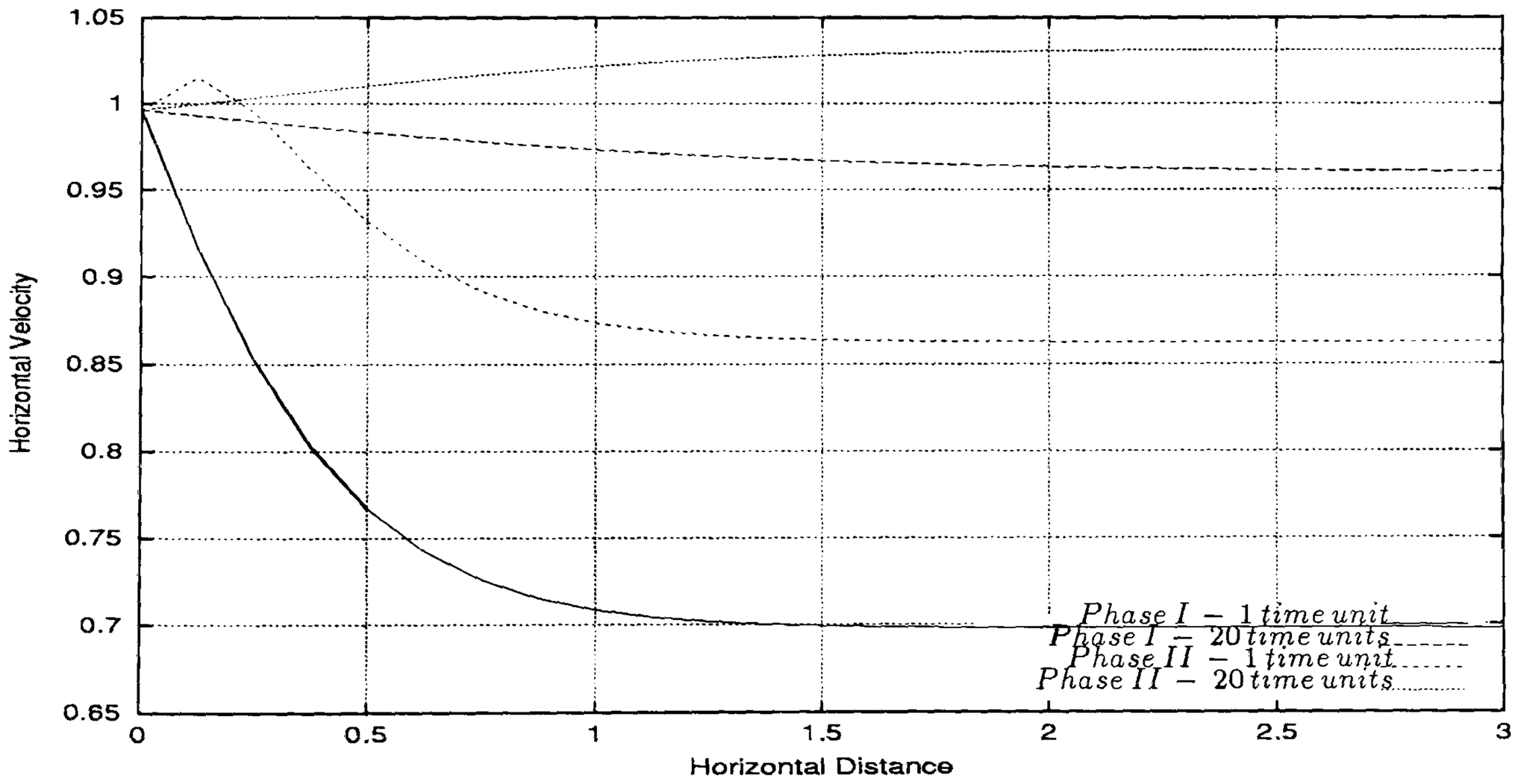


Figure 5.71: Multi-phase channel flow - horizontal velocity profile along the line $y = 0.5$. It can be seen the growing horizontal velocity for both phases from 1 to 20 time units.

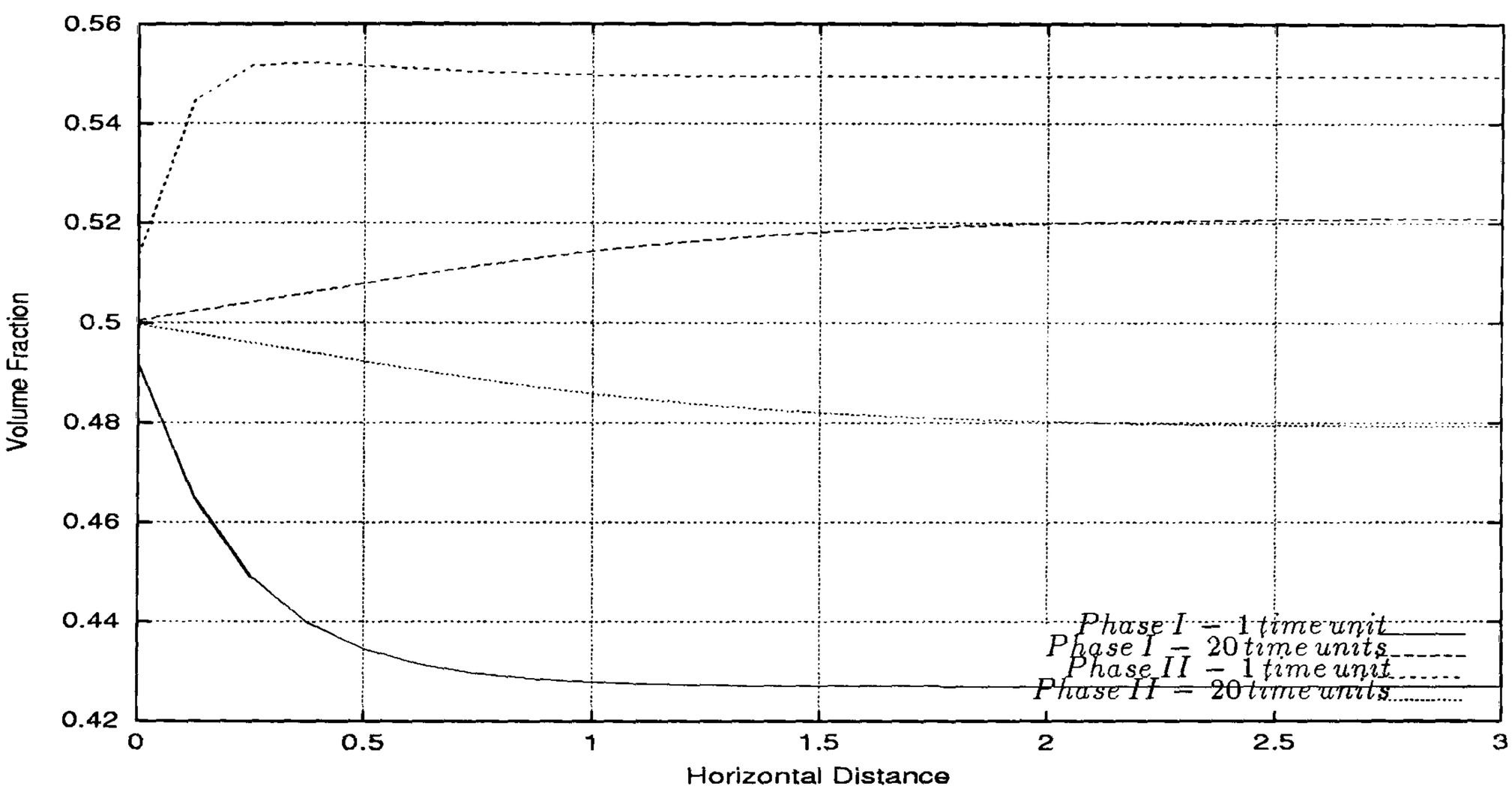


Figure 5.72: Multi-phase channel flow - volume fraction profile along the line $y = 0.5$. It can be verified the approximation of the volume fractions values at 20 time units

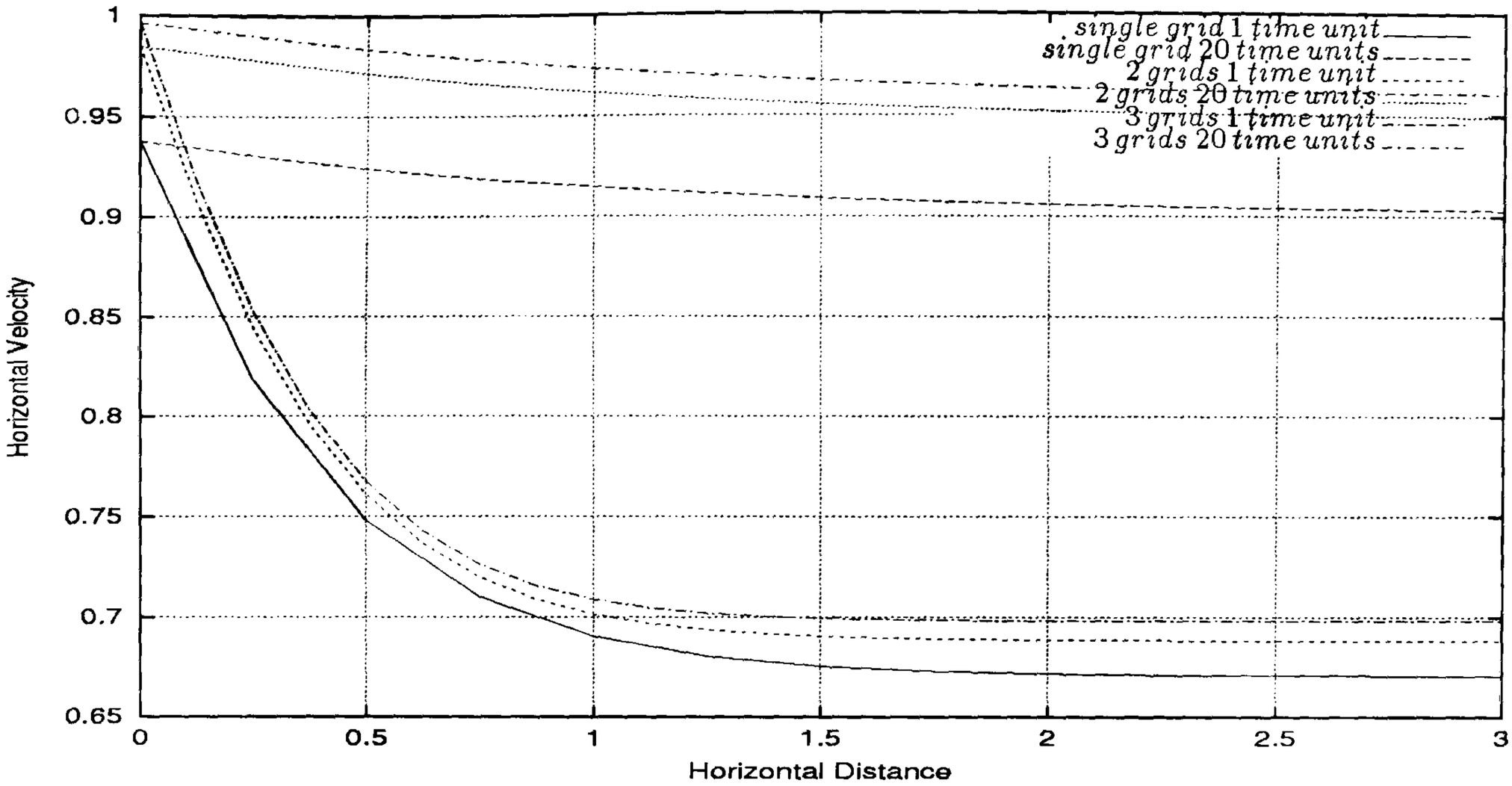


Figure 5.73: Multi-phase channel flow - comparison of horizontal velocity profiles for *phase I* along the line $y = 0.5$ for single 2 and 3 level computations. It can be verified the same growing behaviour from 1 to 20 time units for all level calculations.

Grid and Time Step Independence of the Solutions

In Figures 5.73 through 5.83 are presented the results for the simulation of the channel two-phase flow at 1 and 20 time units for single, 2 and 3 level computations. Good general agreement can be observed in the solutions computed by all the three level calculations.

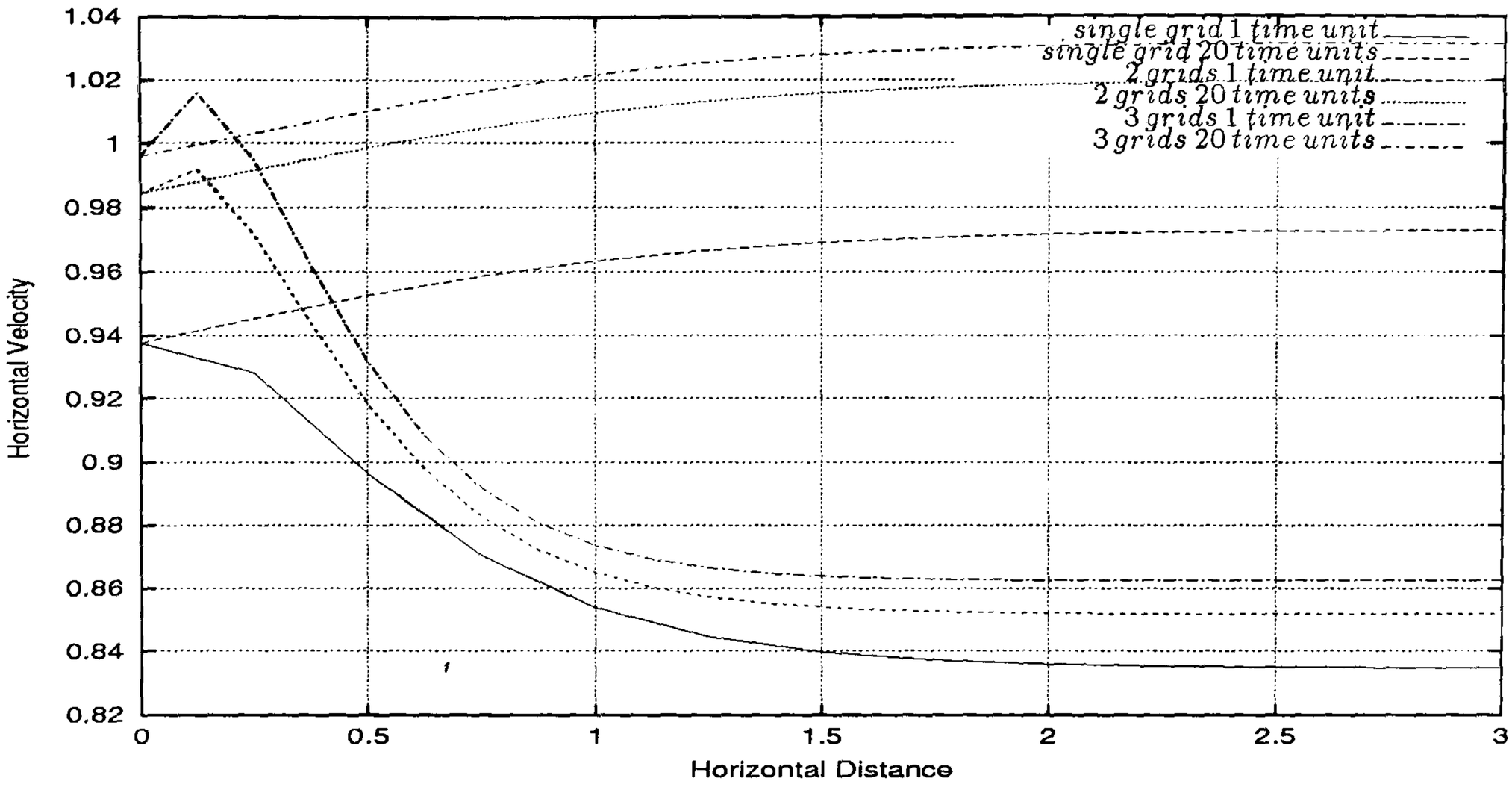


Figure 5.74: Multi-phase channel flow - comparison of horizontal velocity profiles for *phase II* along the line $y = 0.5$ for single 2 and 3 level computations. It can be verified the same growing behaviour from 1 to 20 time units for all level calculations.

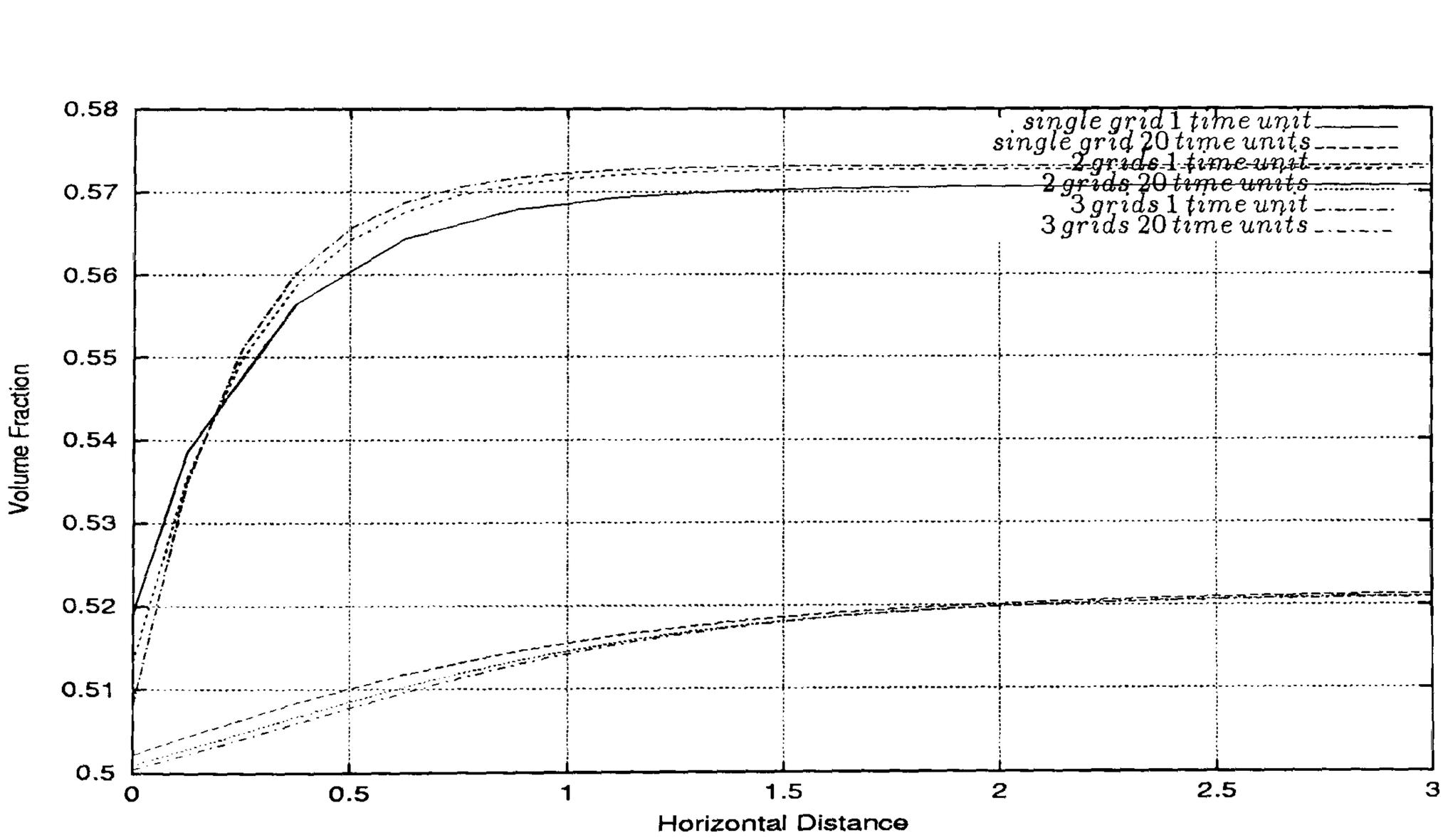


Figure 5.75: Multi-phase channel flow - comparison of volume fraction profiles for *phase I* along the line $y = 0.5$ for single 2 and 3 level computations. It can be seen that the volume fractions approach the value 0.5 for all level calculations at 20 time units.

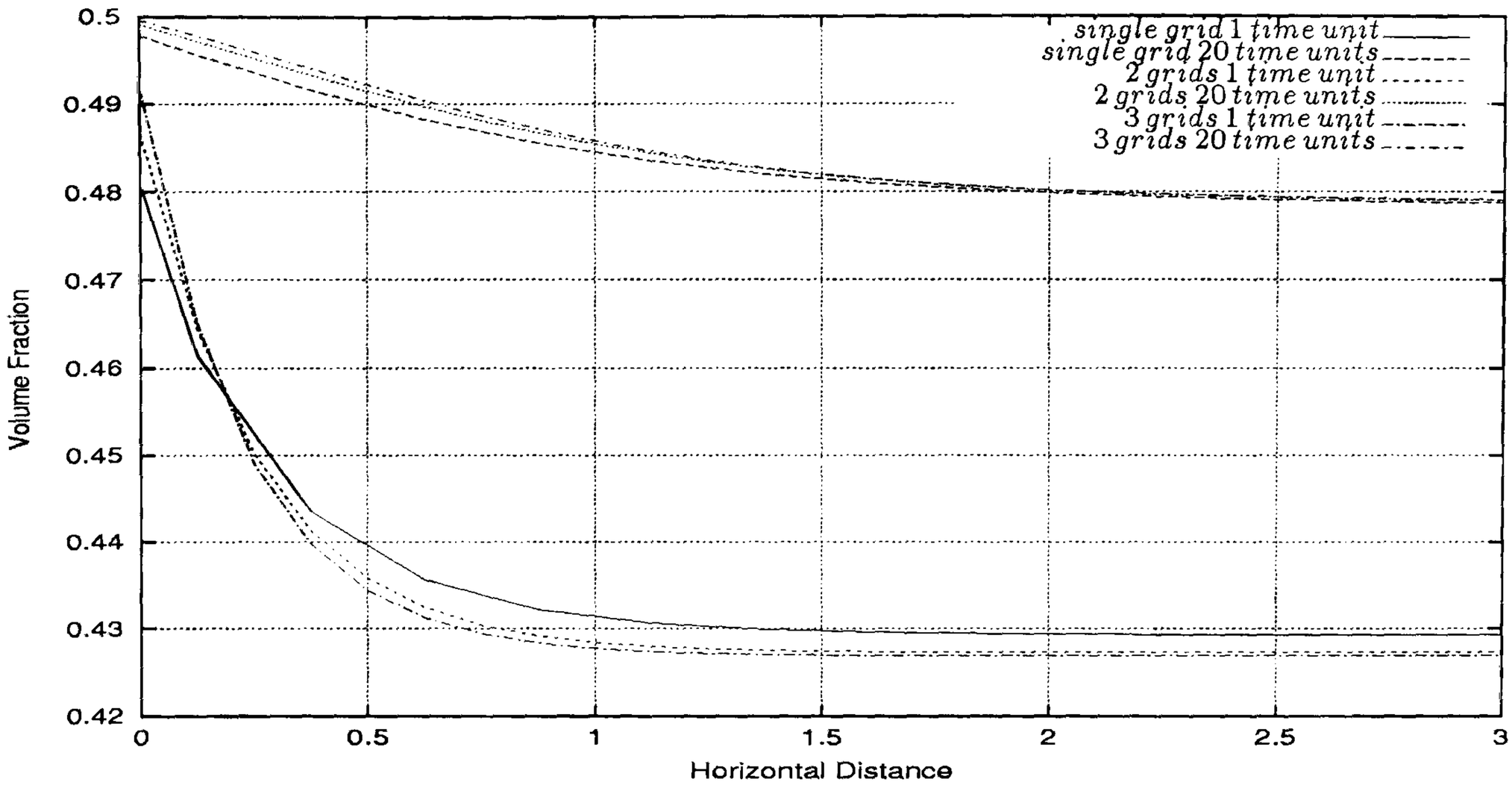


Figure 5.76: Multi-phase channel flow - comparison of volume fraction profiles for *phase II* along the line $y = 0.5$ for single 2 and 3 level computations. It can be seen that the volume fractions approach the value 0.5 for all level calculations at 20 time units.

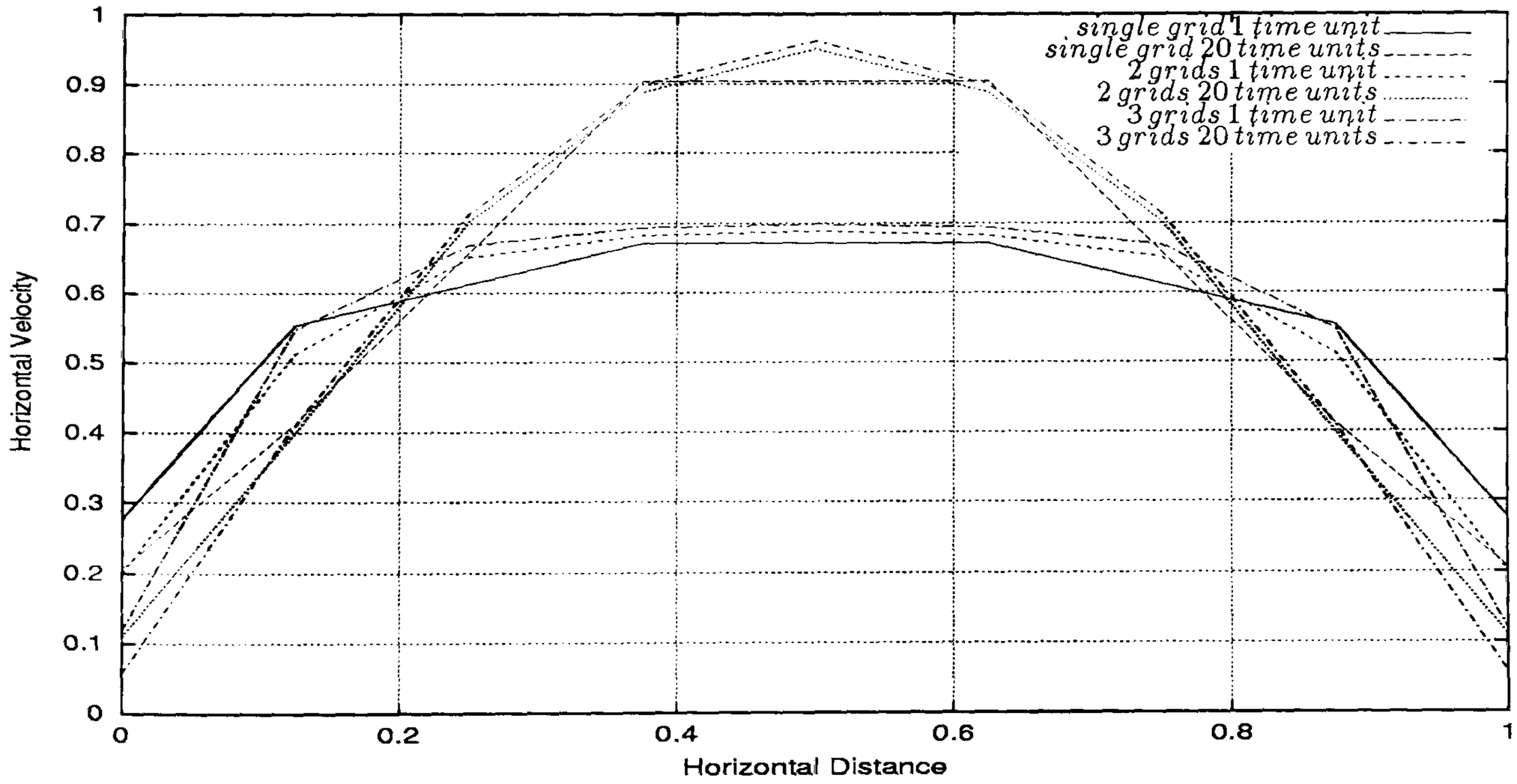


Figure 5.77: Multi-phase channel flow - comparison of horizontal velocity profiles for *phase I* along the line $x = 2.625$ for single 2 and 3 level computations. It can be seen the fully developed parabolic profile for all level calculations at 20 time units.

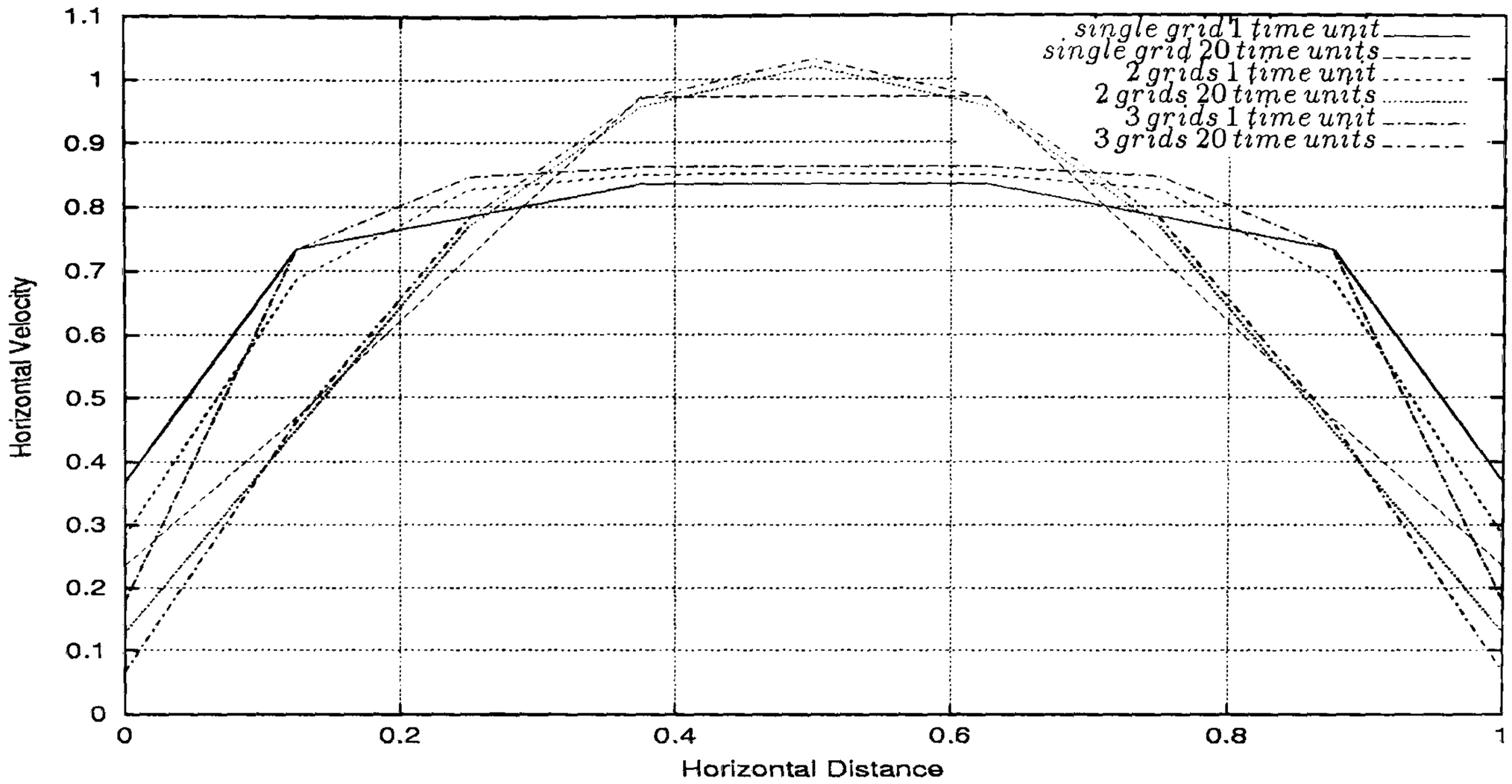


Figure 5.78: Multi-phase channel flow - comparison of horizontal velocity profiles for *phase II* along the line $x = 2.625$ for single 2 and 3 level computations. It can be observed the fully developed parabolic profile for all level calculations at 20 time units.

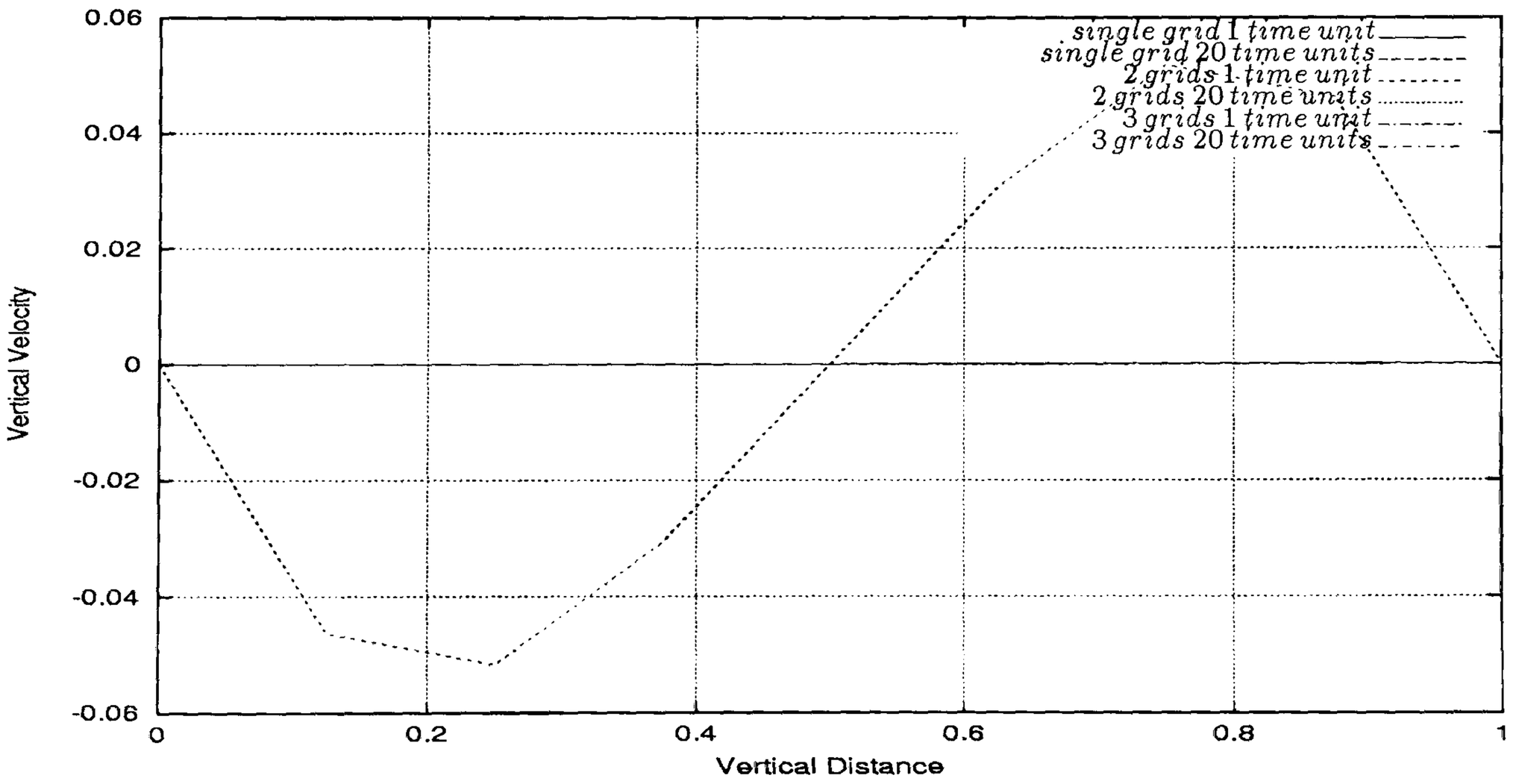


Figure 5.79: Multi-phase channel flow - comparison of vertical velocity profiles for *phase I* along the line $x = 2.625$ for single 2 and 3 level computations. It can be verified good agreement between all level calculations at 1 and 20 time units.

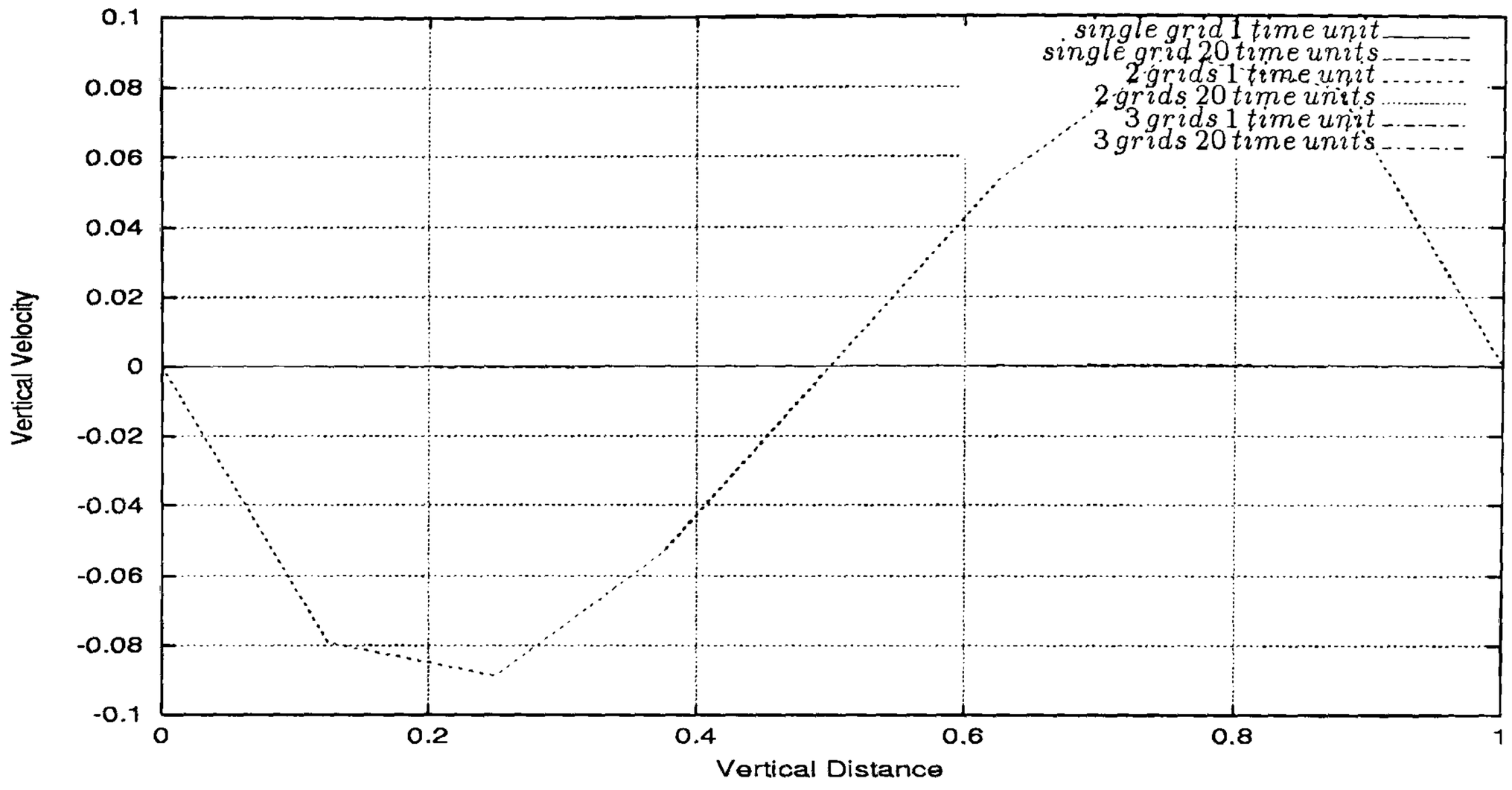


Figure 5.80: Multi-phase channel flow - comparison of vertical velocity profiles for *phase II* along the line $x = 2.625$ for single 2 and 3 level computations. It can be verified good agreement between all level calculations at 1 and 20 time units.

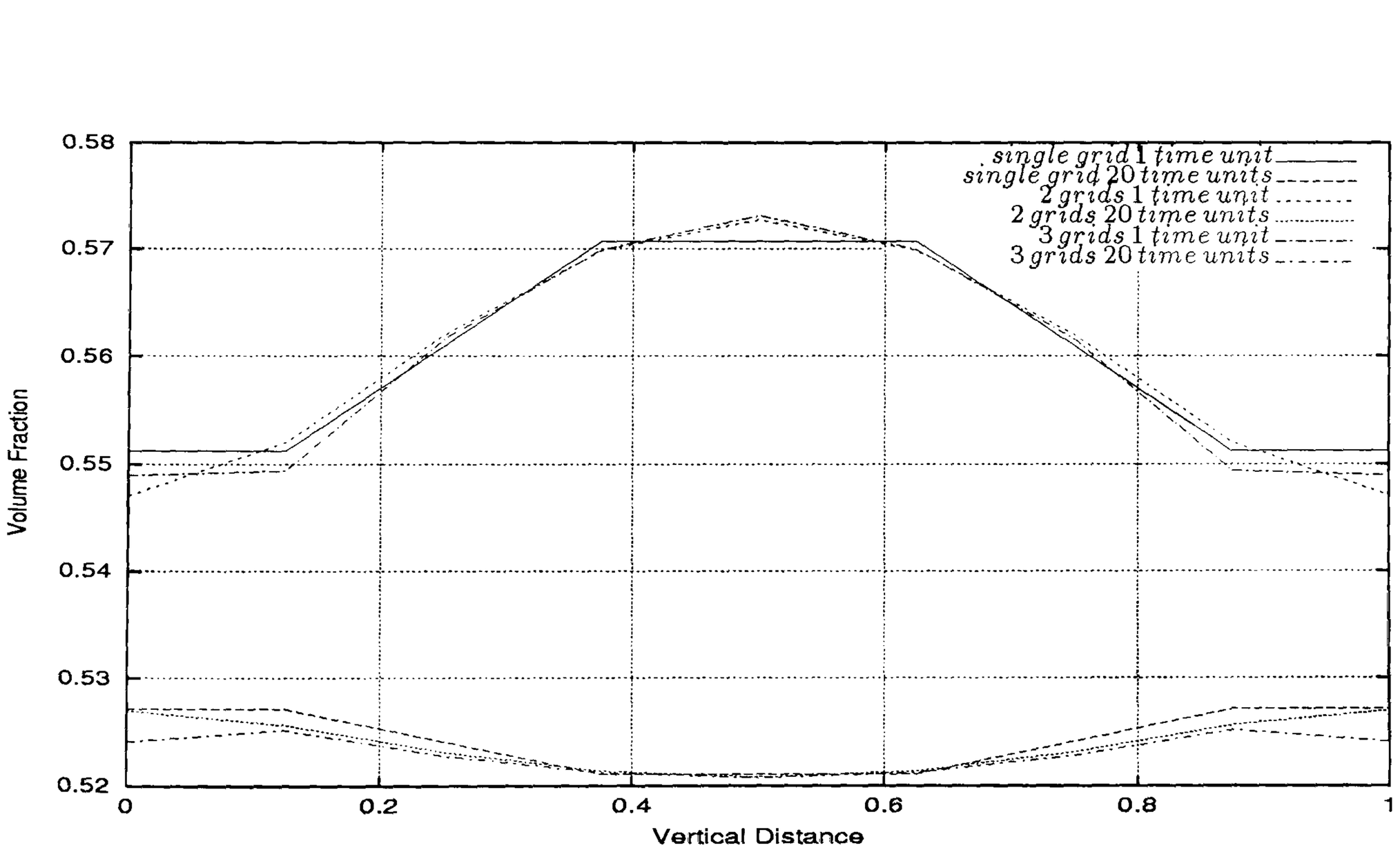


Figure 5.81: Multi-phase channel flow - comparison of volume fraction profiles for *phase I* along the line $x = 2.625$ for single 2 and 3 level computations. It can be verified the good agreement between all level calculations at 1 and 20 time units.

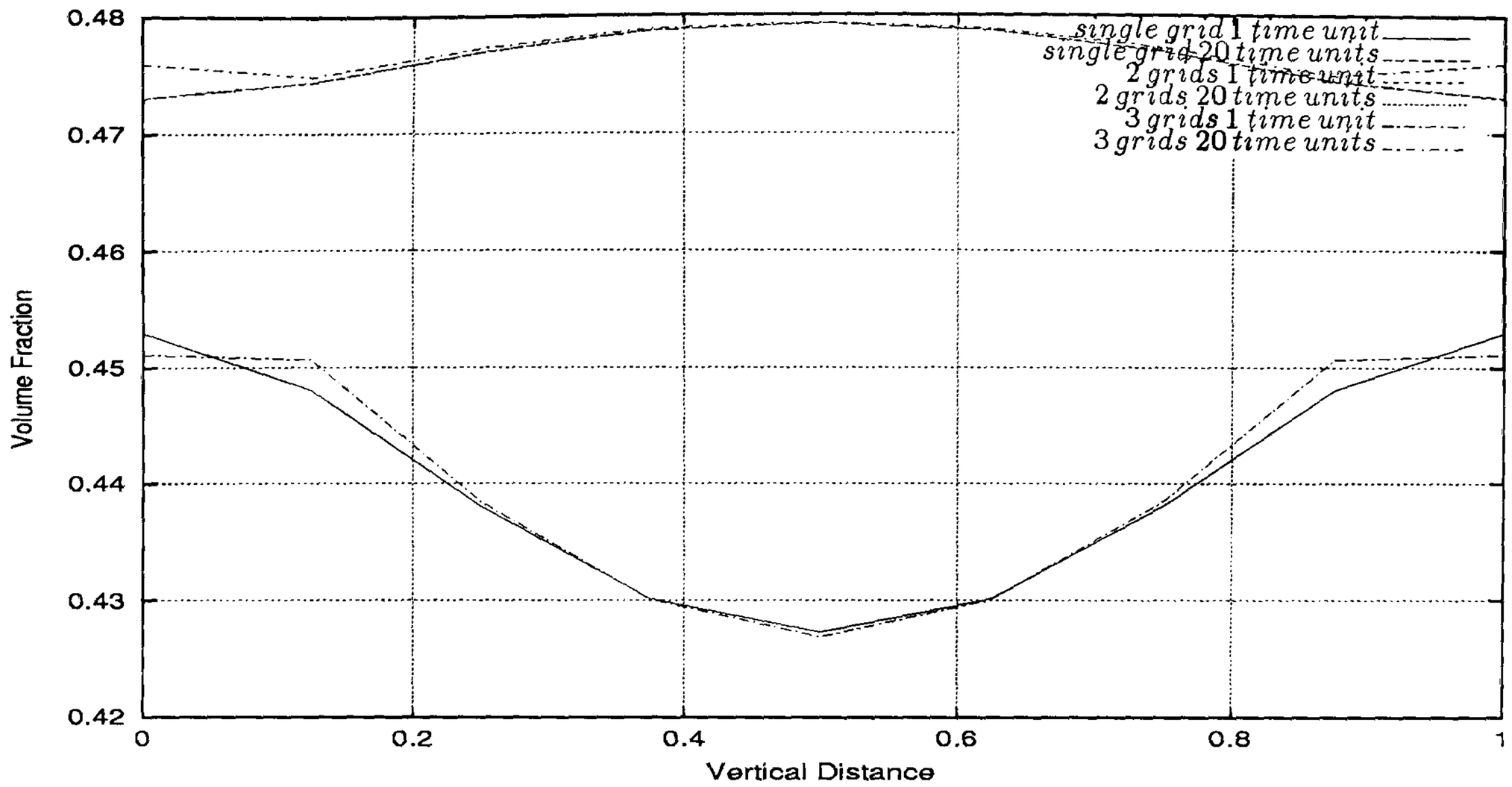


Figure 5.82: Multi-phase channel flow - comparison of volume fraction profiles for *phase II* along the line $x = 2.625$ for single 2 and 3 level computations. It can be observed the good agreement between all level calculations at 1 and 20 time units.

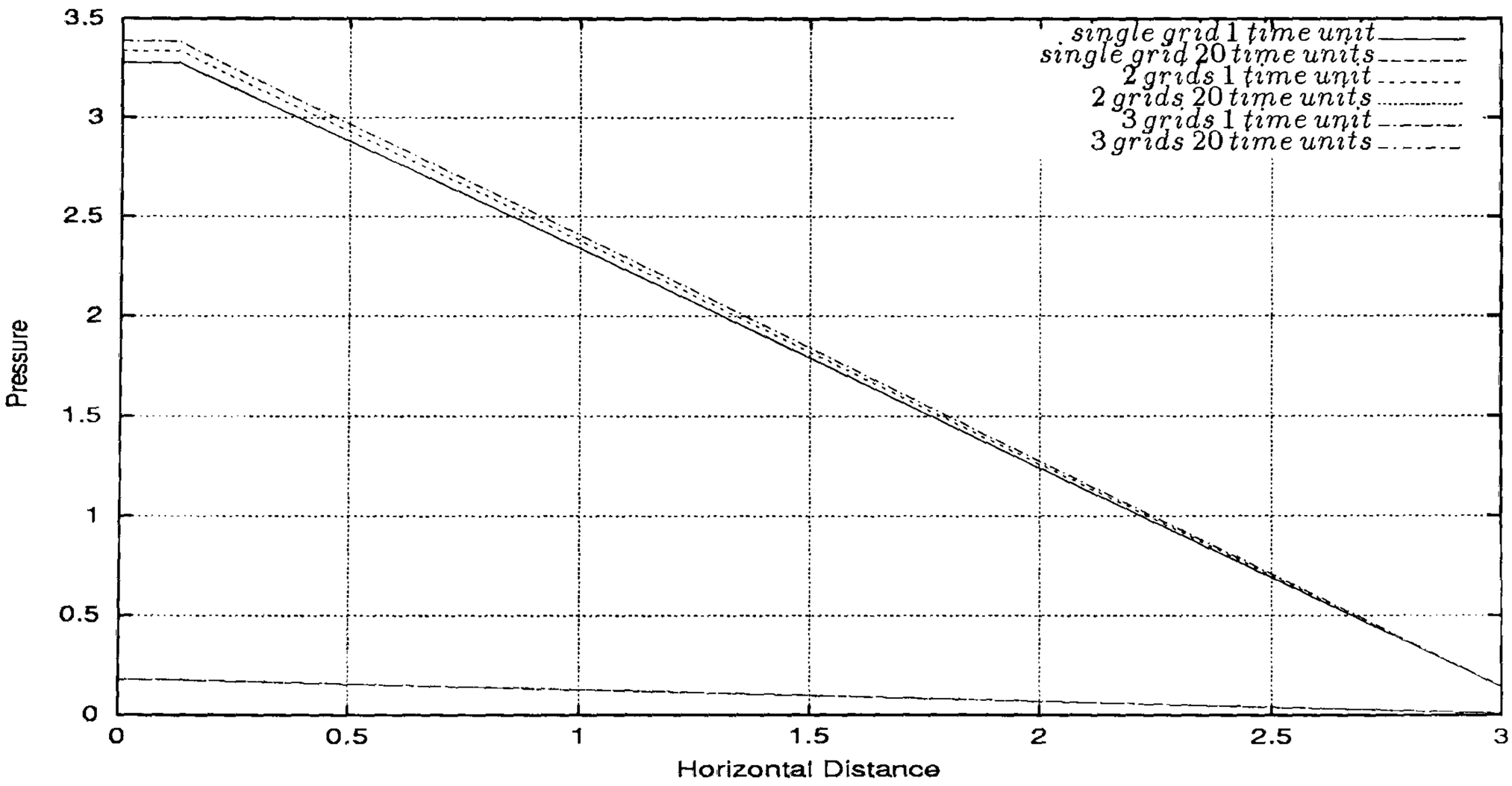


Figure 5.83: Multi-phase channel flow - comparison of pressure drop profiles along the line $y = 0.5$ for single 2 and 3 level computations. It can be verified the good agreement between the pressure drop profiles produced by all level calculations at 1 and 20 time units.

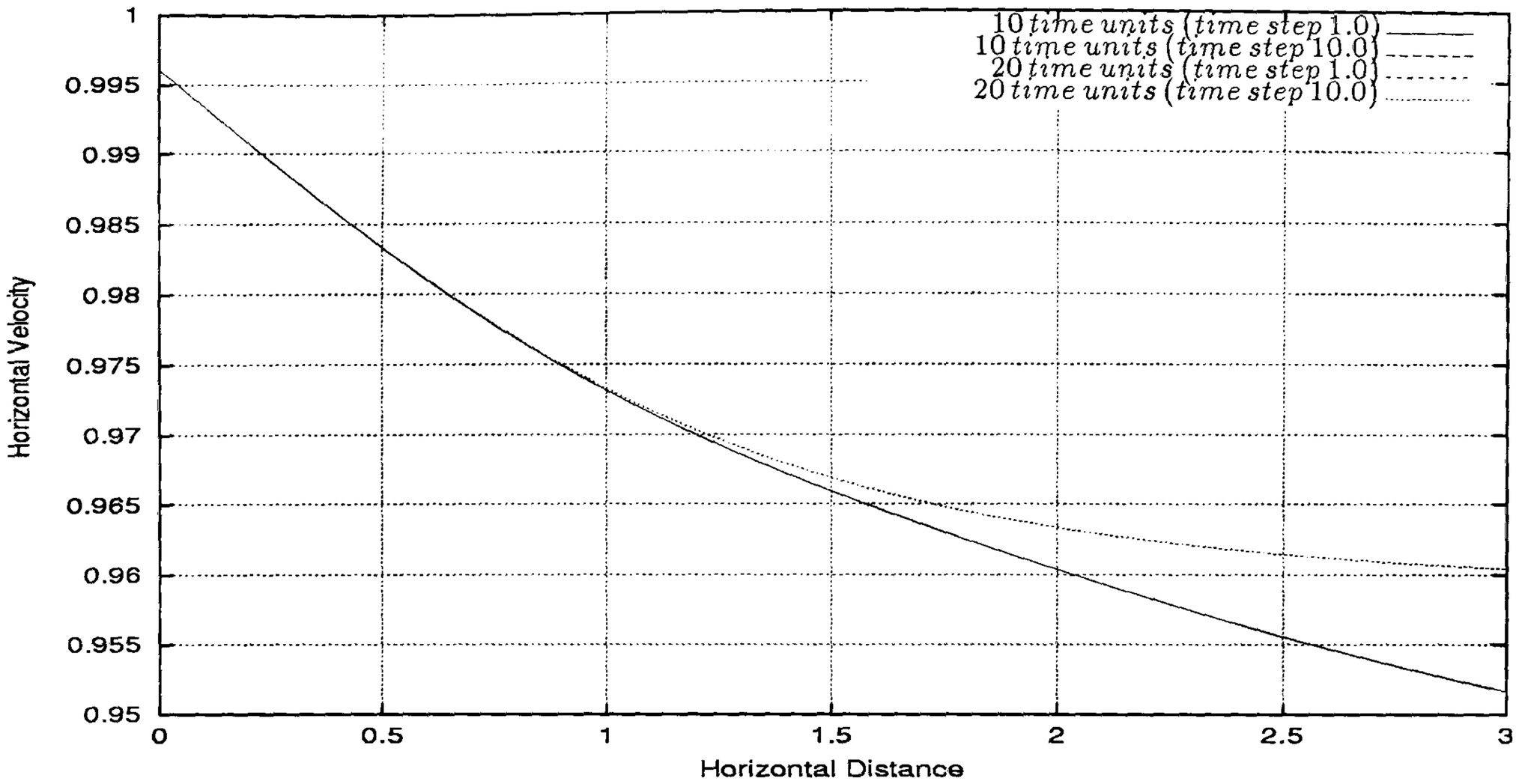


Figure 5.84: Multi-phase channel flow - comparison of horizontal velocity profiles for *phase I* along the line $y = 0.5$ for 3 level computations and different time steps. It can be observed good agreement between the profiles at 10 and 20 time units

Figures 5.84 through 5.94 show the results for this test case at 10 and 20 time units for a 3 level simulation with time steps 1.0 and 10.0. In Figures 5.84 and 5.85 it can be seen that there is good agreement between the horizontal velocities along the line $y = 0.5$ at 10 and 20 time units for both phases. The horizontal velocities along the line $x = 2.625$, (Figures 5.88 and 5.89) exhibit good agreement particularly when the time is 20 time units. We can conclude that good agreement is observed between the horizontal velocity profiles. Figures 5.86 and 5.87 show the volume fractions profile along the line $y = 0.5$. Some disagreement is observed particularly when time is 10 time units. However at 20 time units the two curves are very close. This behaviour is also observed in Figures 5.92 and 5.93 which represent the volume fractions profile along the line $x = 2.625$. Hence the volume fractions profiles show some disagreement. This suggests that the volume fractions are less time step independent than the velocities. Furthermore, as the simulation advance on time the agreement between the profiles are better. Figure 5.94, which represent the pressure drop along the line $y = 0.5$ suggests this.

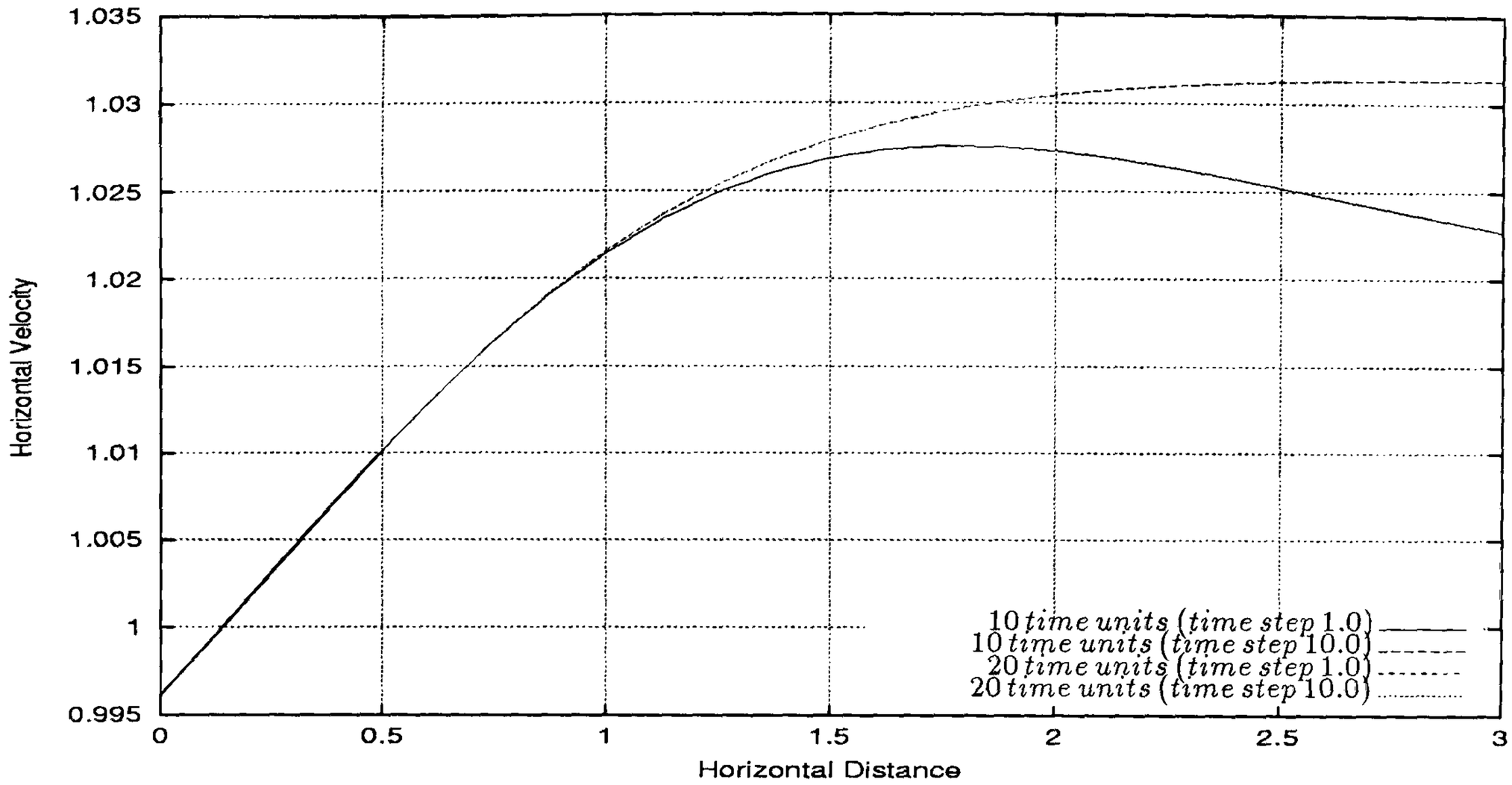


Figure 5.85: Multi-phase channel flow - comparison of horizontal velocity profiles for *phase II* along the line $y = 0.5$ for 3 level computations and different time steps. It can be seen good agreement between the profiles at 10 and 20 time units.

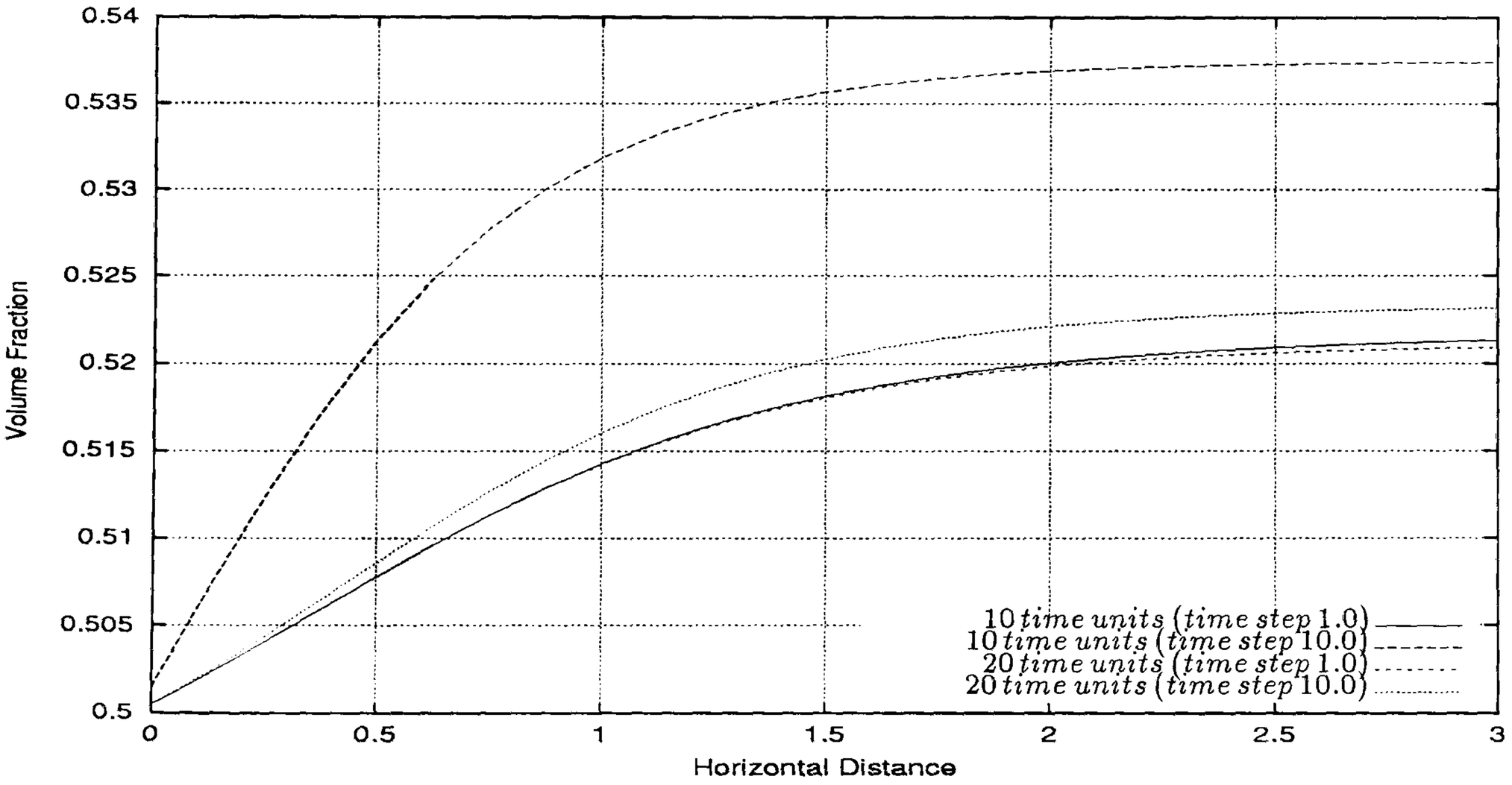


Figure 5.86: Multi-phase channel flow - comparison of volume fraction profiles for *phase I* along the line $y = 0.5$ for 3 level computations and different time steps. It can be observed good agreement between the volume fractions profiles at 20 time units. Note the small scale of the volume fraction axis.

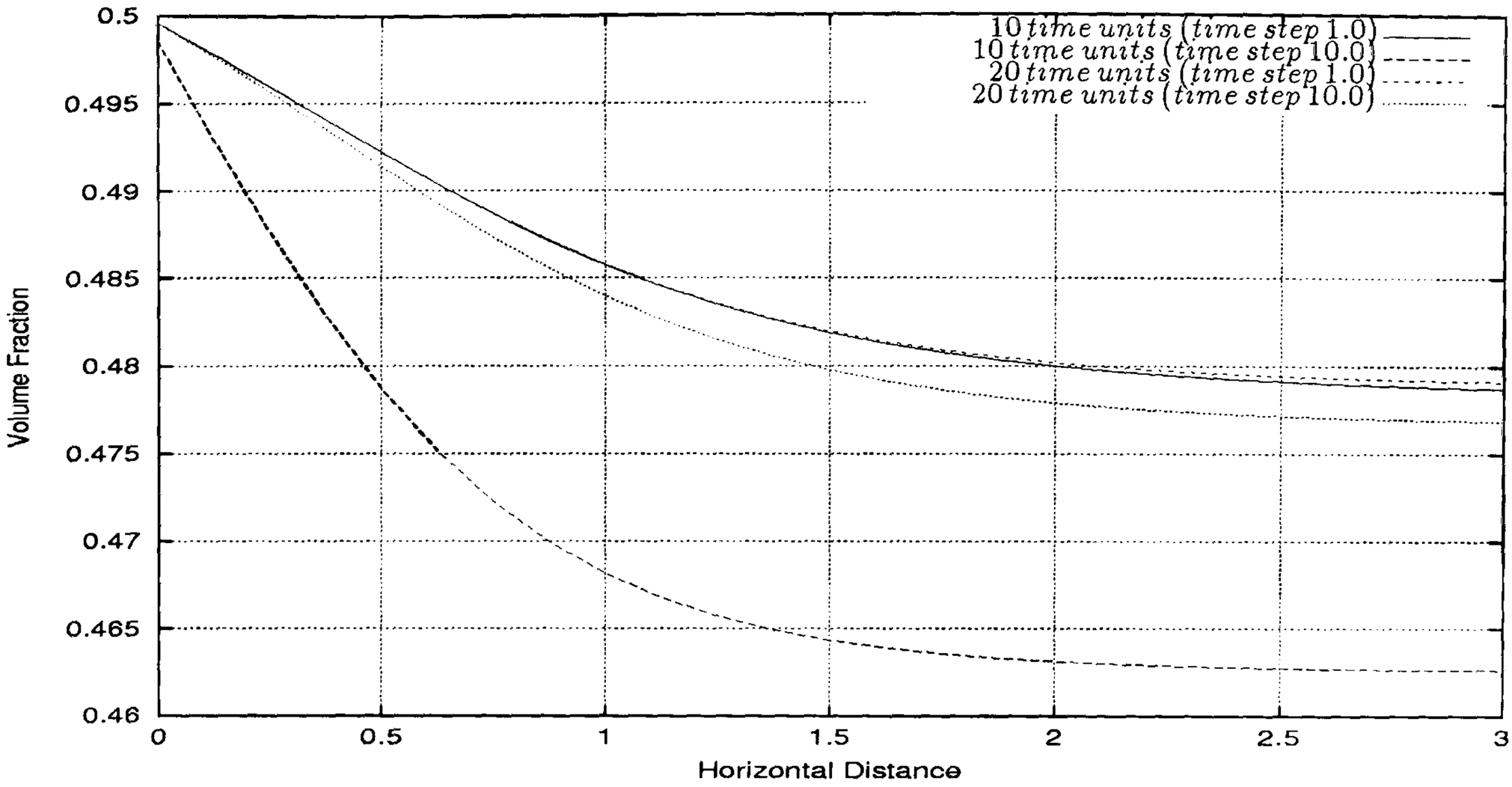


Figure 5.87: Multi-phase channel flow - comparison of volume fraction profiles for *phase II* along the line $y = 0.5$ for 3 level computations and different time steps. It can be seen good agreement between the volume fractions profiles at 20 time units.

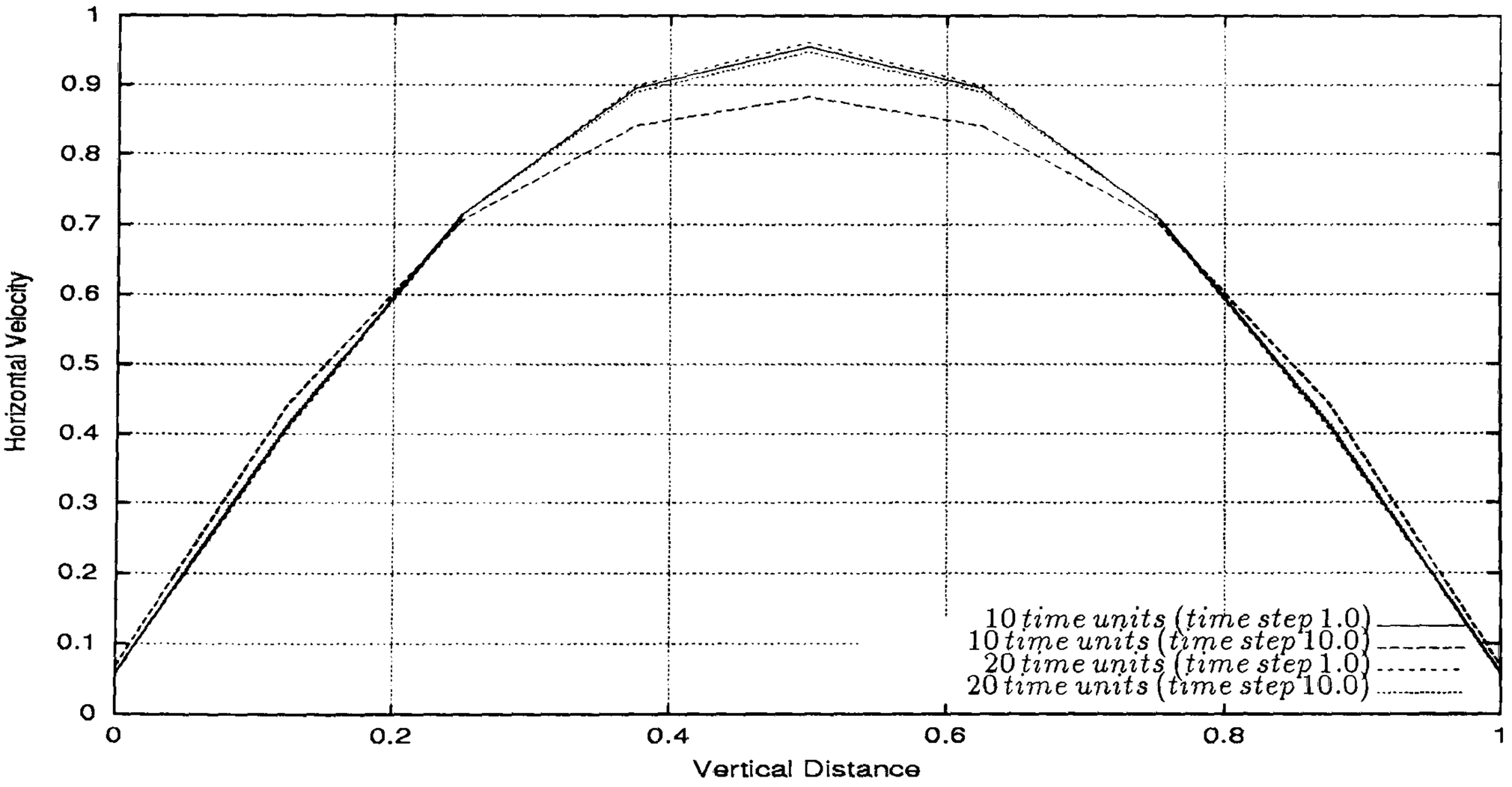


Figure 5.88: Multi-phase channel flow - comparison of horizontal velocity profiles for *phase I* along the line $x = 2.625$ for 3 level computations and different time steps. It can be observed good agreement in the fully developed parabolic profiles at 20 time units.

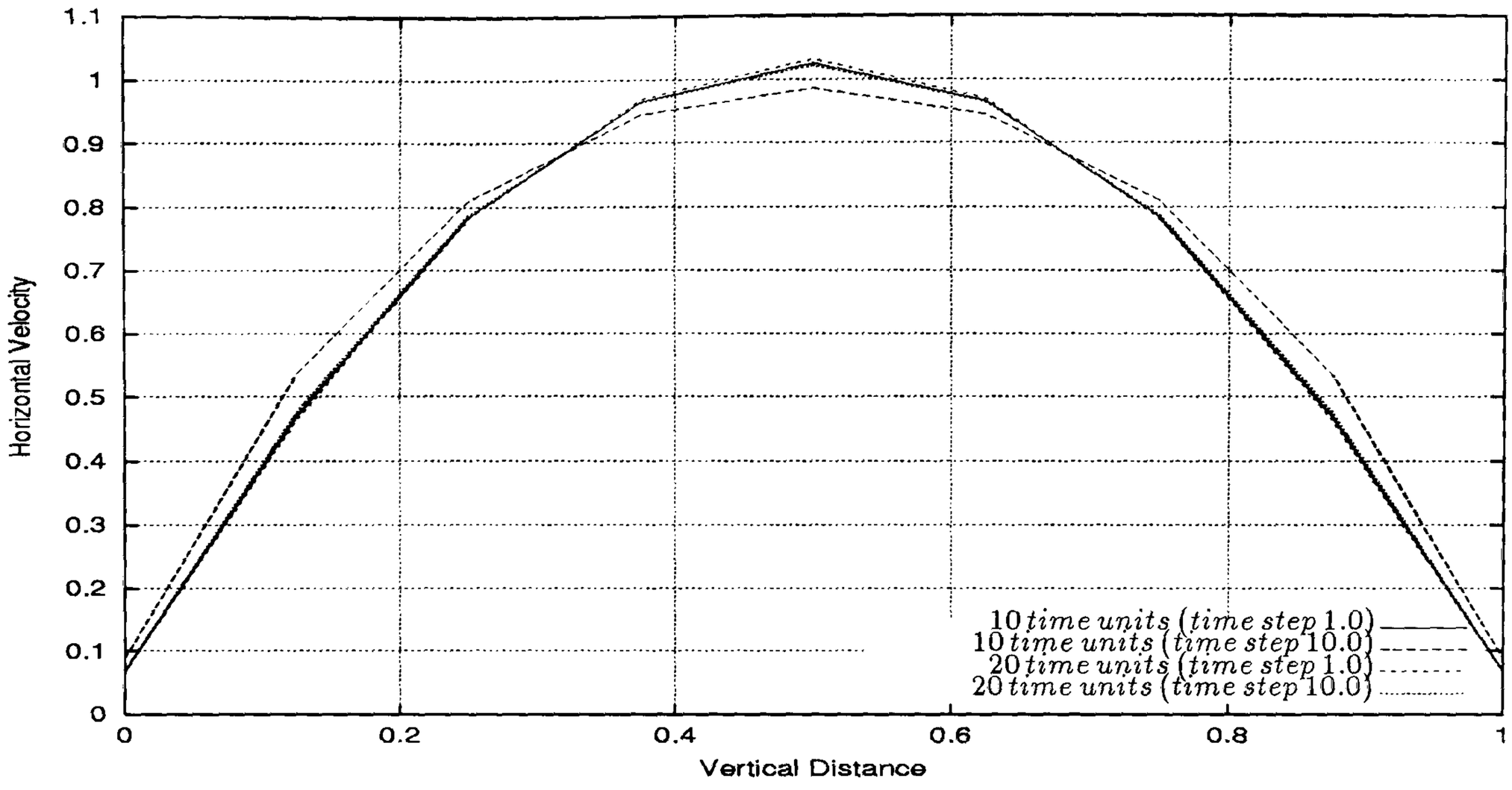


Figure 5.89: Multi-phase channel flow - comparison of horizontal velocity profiles for *phase II* along the line $x = 2.625$ for 3 level computations and different time steps. It can be seen good agreement in the fully developed parabolic profiles at 20 time units.

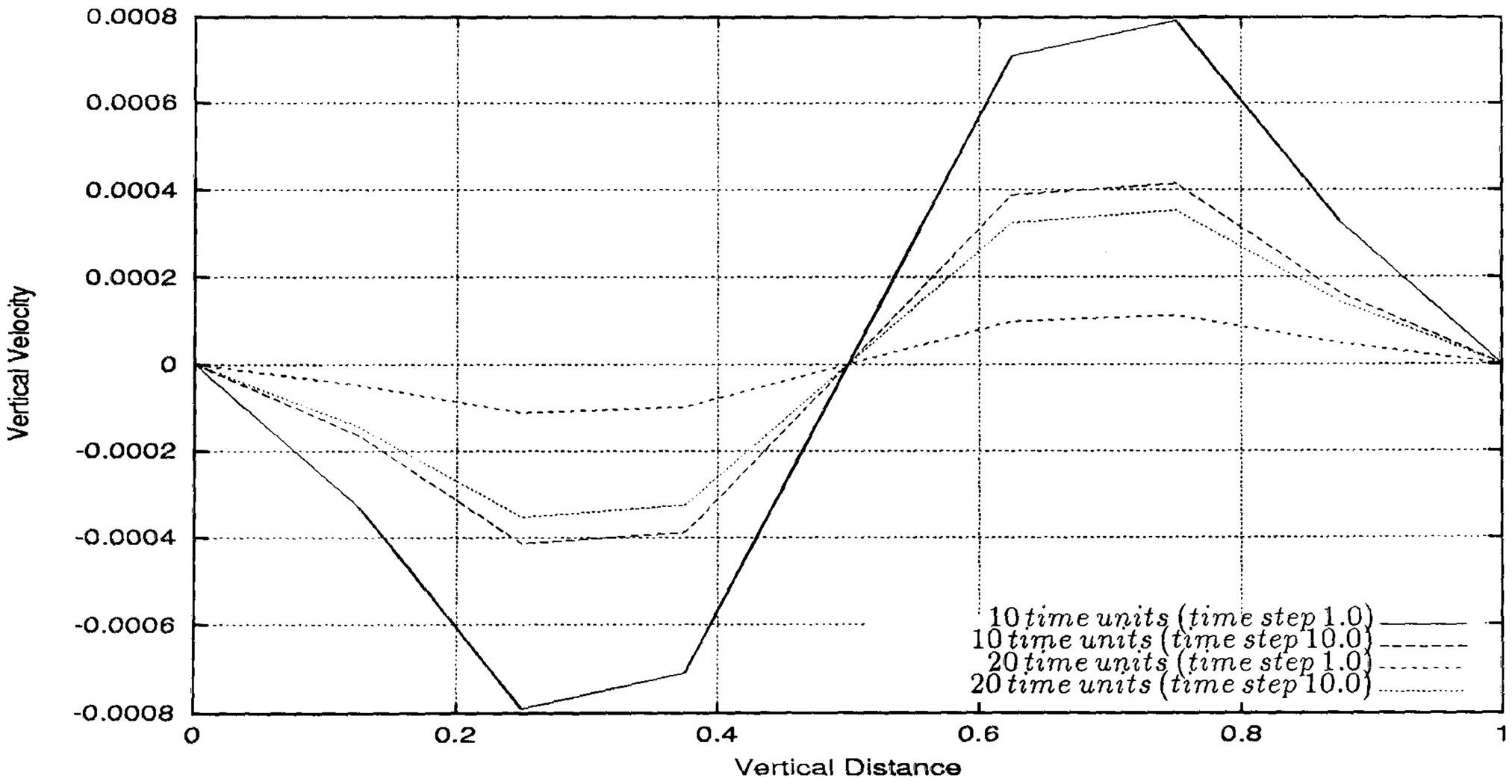


Figure 5.90: Multi-phase channel flow - comparison of vertical velocity profiles for *phase I* along the line $x = 2.625$ for 3 level computations and different time steps. Note the very small values of the vertical velocities at 10 and 20 time units for all the time step computations.

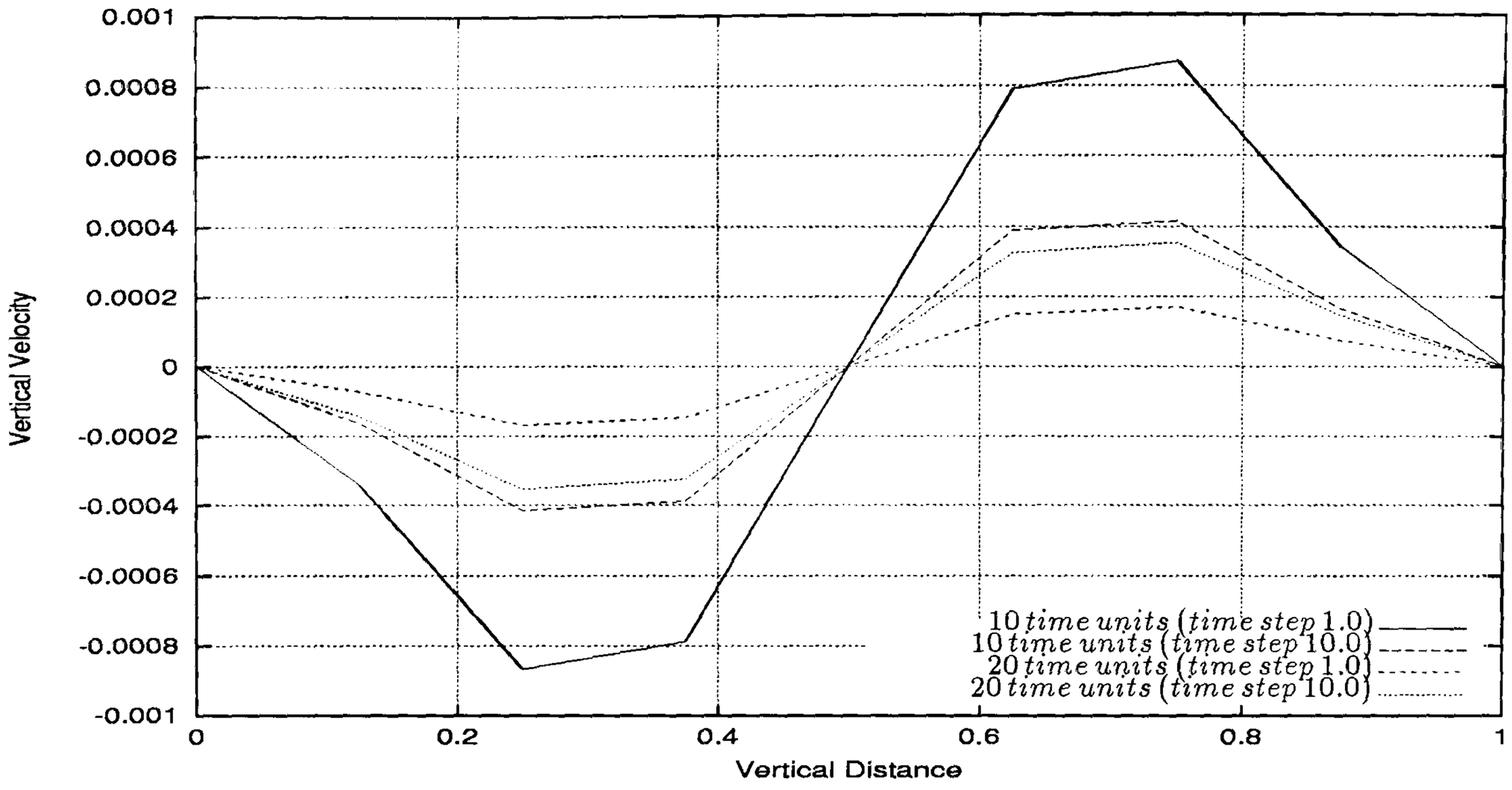


Figure 5.91: Multi-phase channel flow - comparison of vertical velocity profiles for *phase II* along the line $x = 2.625$ for 3 level computations and different time steps. Observe the very small values of the vertical velocities at 10 and 20 time units for all the time step computations.

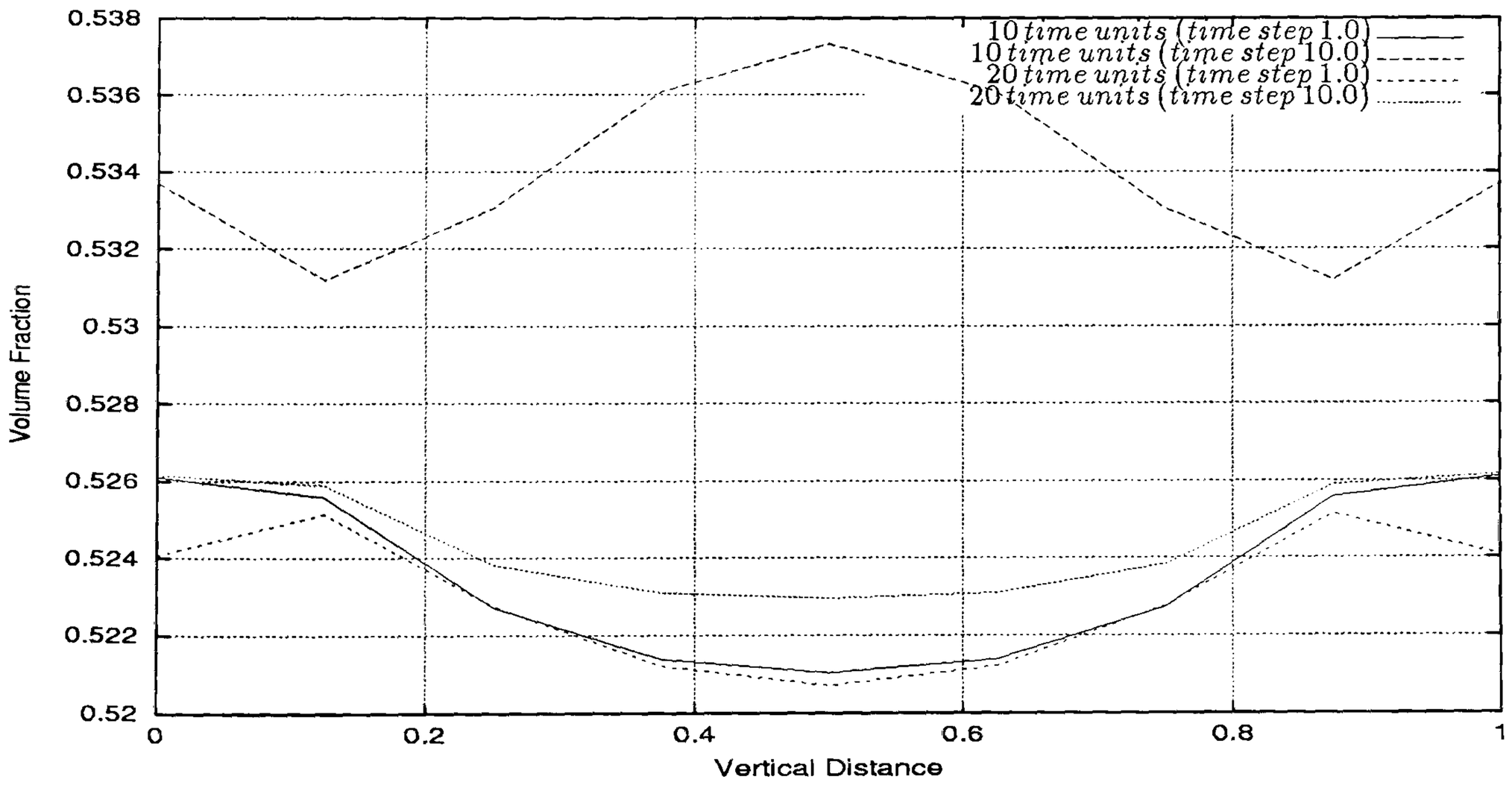


Figure 5.92: Multi-phase channel flow - comparison of volume fraction profiles for *phase I* along the line $x = 2.625$ for 3 level computations and different time steps. It can be seen good agreement between the volume fractions profiles at 20 time units.

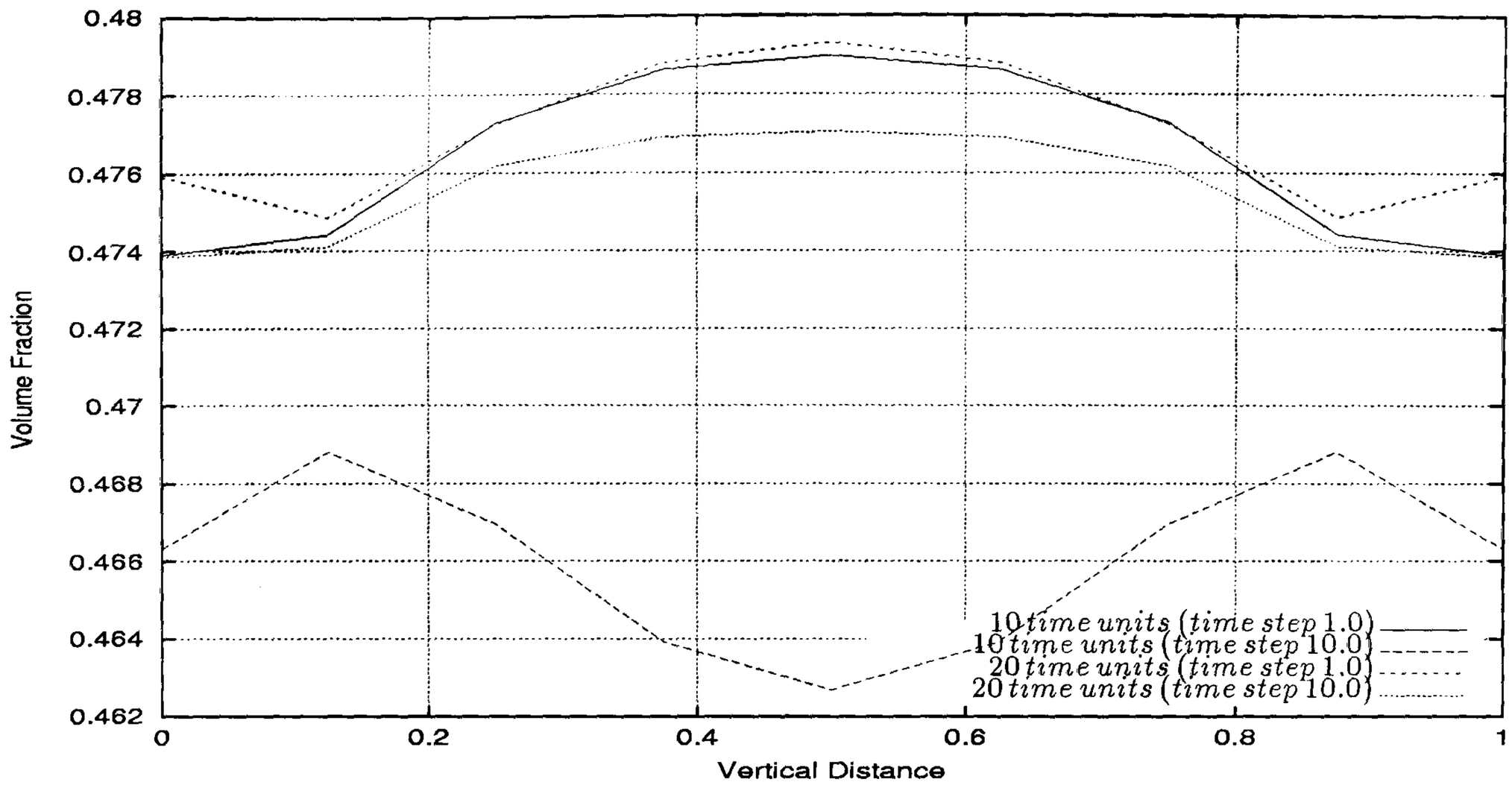


Figure 5.93: Multi-phase channel flow - comparison of volume fraction profiles for *phase II* along the line $x = 2.625$ for 3 level computations and different time steps. It can be observed good agreement between the volume fractions profiles at 20 time units.

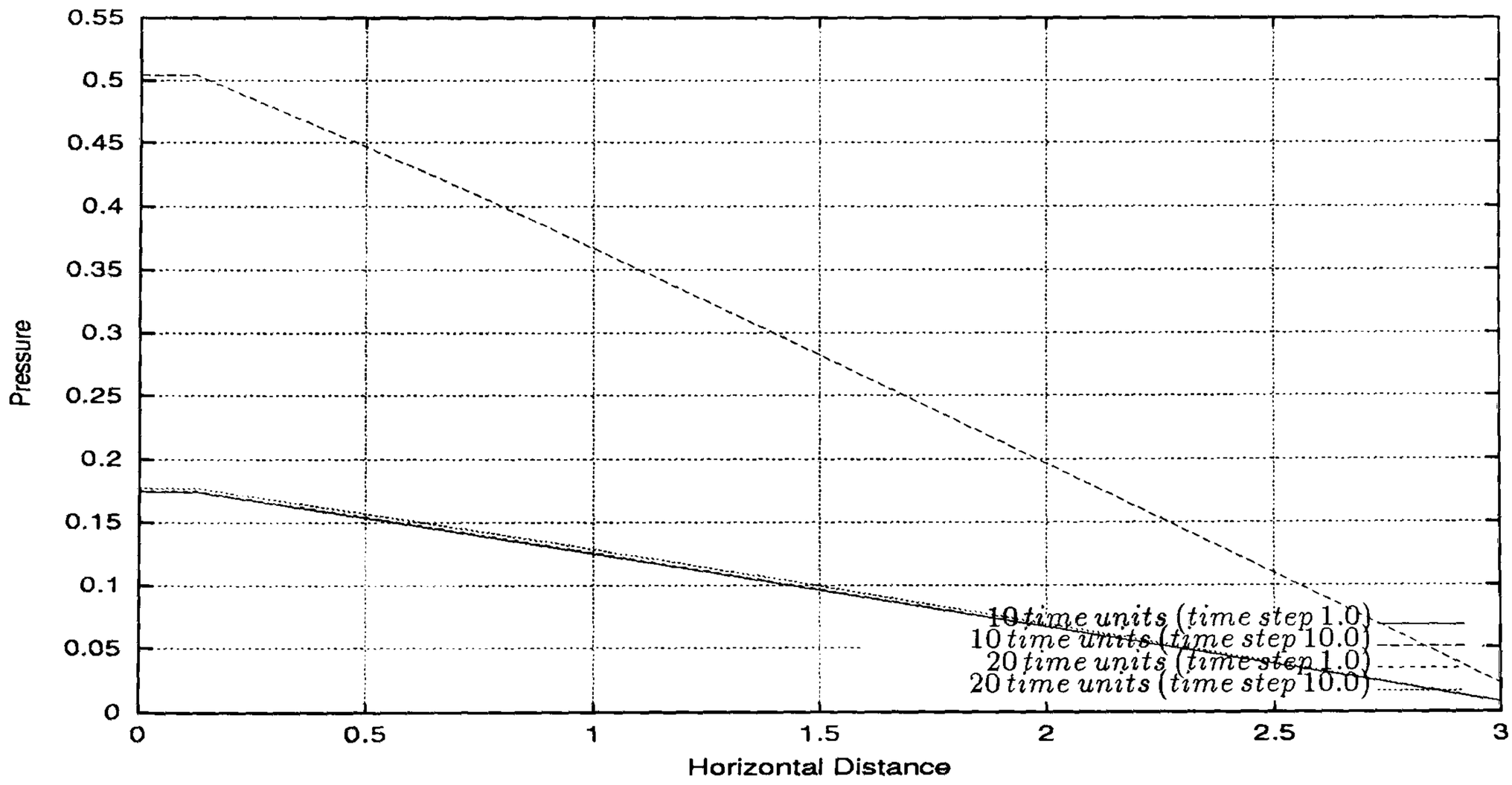


Figure 5.94: Multi-phase channel flow - comparison of pressure drop profiles along the line $y = 0.5$ for 3 level computations and different time steps. It can be observed good agreement between the pressure drop profiles at 20 time units.

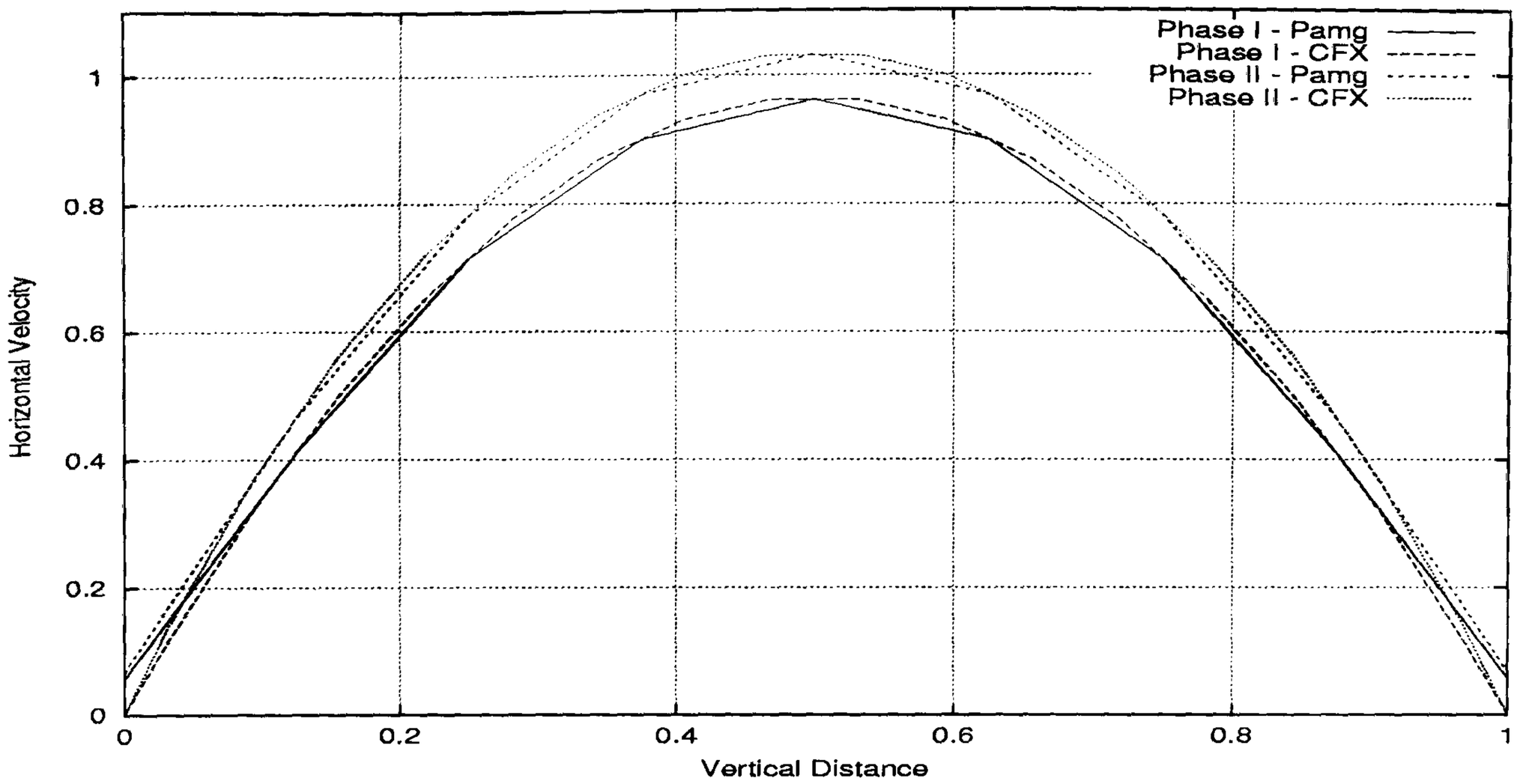


Figure 5.95: Multi-phase channel flow - horizontal velocity profile along the line $x = 2.625$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be observed good agreement between the parabolic profiles produced by both solvers.

Comparison with `CFX 4.3` Solutions

Figures 5.95 through 5.99 present the comparison between the `transient Pamg Multi-phase` and `CFX 4.3` solvers based on the horizontal velocity, vertical velocity, volume fraction and pressure drop along the lines $x = 2.625$ and $y = 0.5$ at the integration time of 20 time units. The disagreement that is seen in the volume fractions profile along the line $x = 2.625$ in the walls ($x = 0.0$ and $x = 1.0$) is due to the different volume fractions initial conditions. `CFX 4.3` starts the simulation at each time step fixing the initial boundary conditions for the volume fractions in the walls as 0.5.

Summary

To start the validation process of the `transient Pamg Multi-phase` we chose a very simple problem. It is the simulation of a two-phase fluid flow through a channel. The geometry chosen, for a question of simplicity has been exactly identical to the geometry of the Single-phase channel flow. Two-different fluids with different viscosities and densities have been used in this simulation. Results in a 3 level computation have been presented. Additionally they have been compared with the results produced by the solver `CFX 4.3` in the same geometry domain. Generally we observe very good agreement between the results computed by both solvers. However the volume fractions near the walls do not match exactly. This is due to different initialization conditions for the volume fractions in the `transient Pamg Multi-phase` and `CFX 4.3`.

The grid and time step independence of the solutions has been addressed in this test case. The results verified for single, 2 and 3 level computations agree

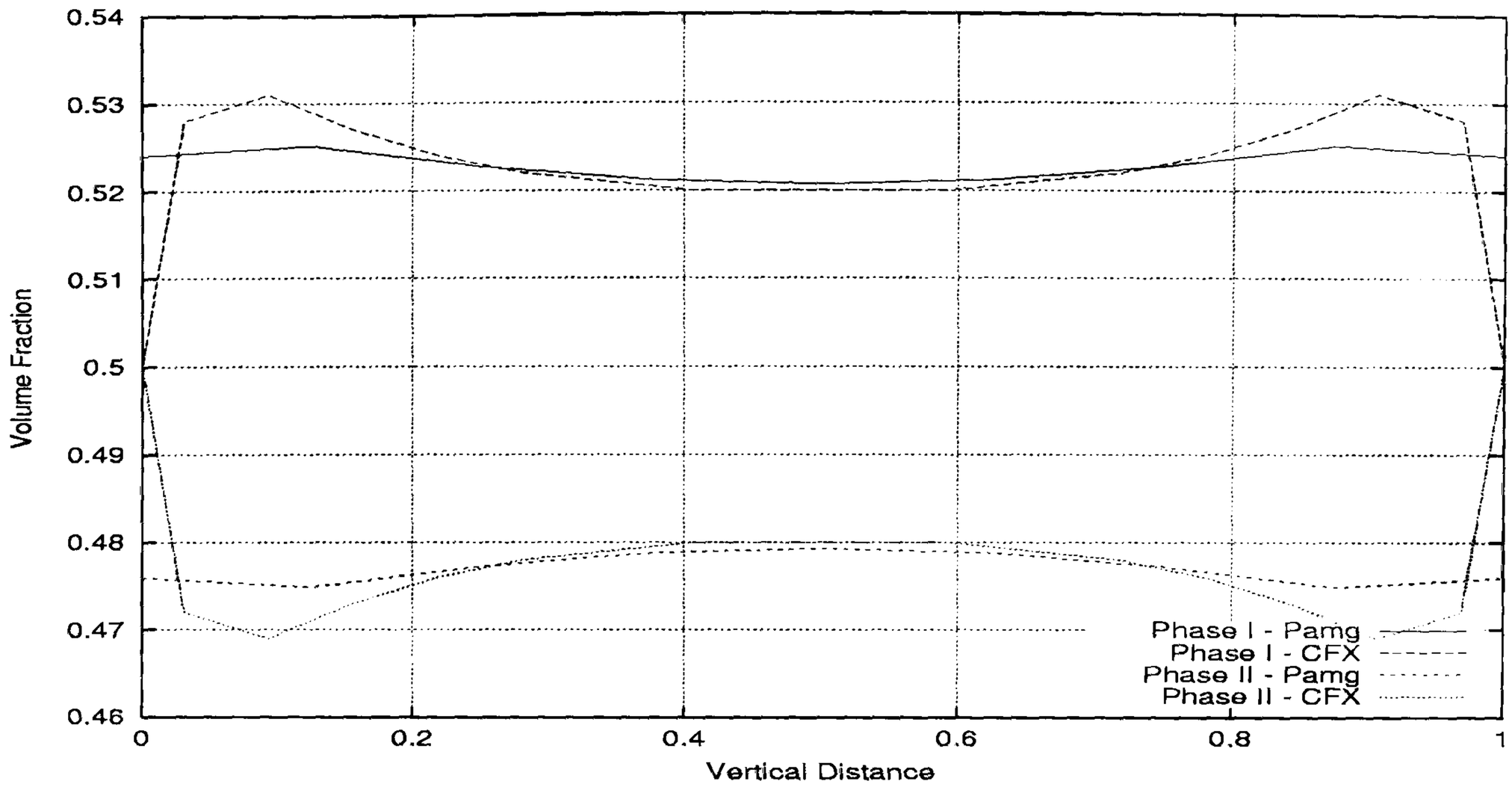


Figure 5.96: Multi-phase channel flow - volume fraction profile along the line $x = 2.625$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen the different volume fractions conditions in the walls used by both solvers.

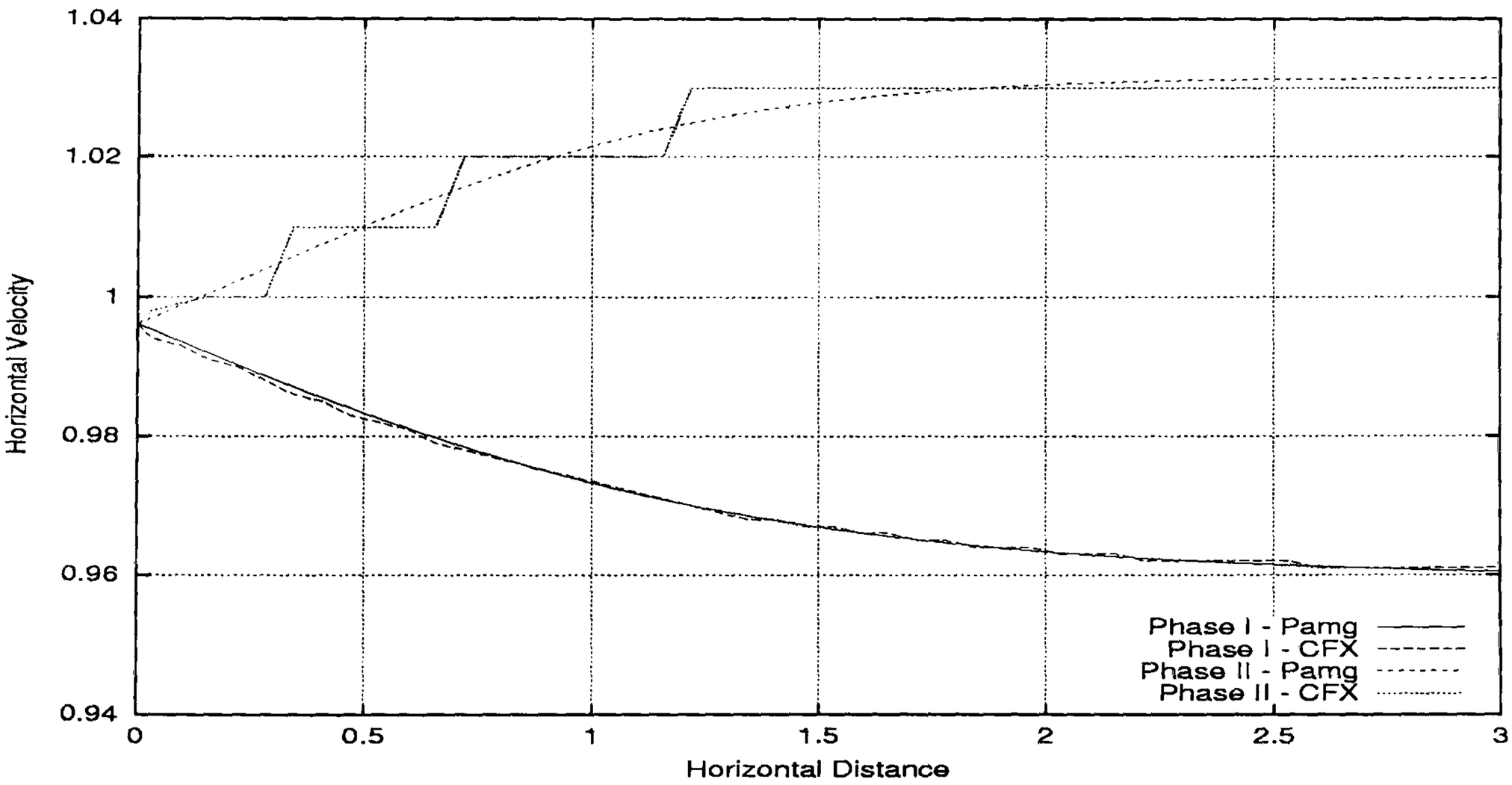


Figure 5.97: Multi-phase channel flow - horizontal velocity profile along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be observed good agreement between the results produced by both solvers. Note the little jumps that `CFX 4.3` produces for *phase II*.

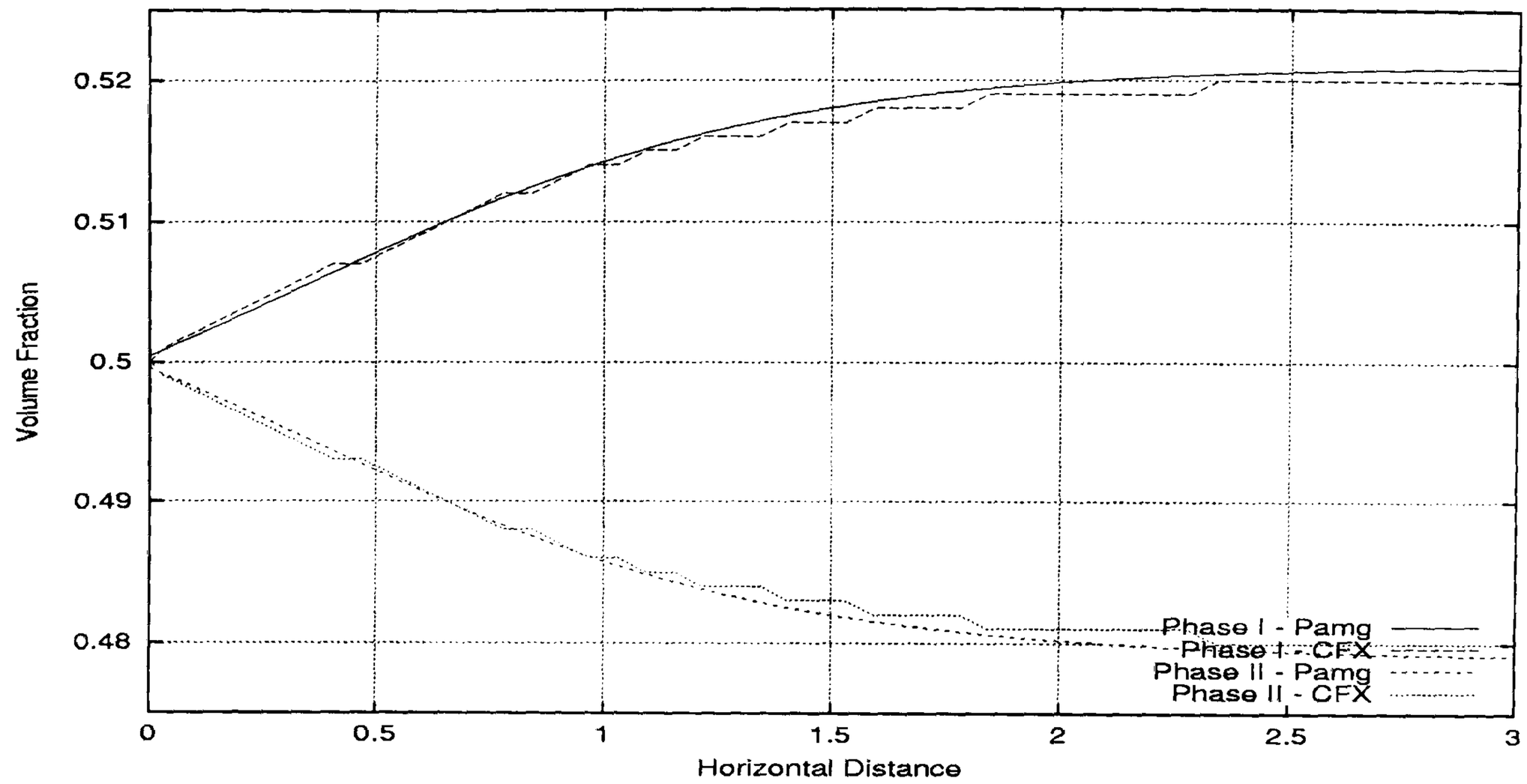


Figure 5.98: Multi-phase channel flow - volume fraction profile along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen good agreement between volume fractions produced by both solvers.

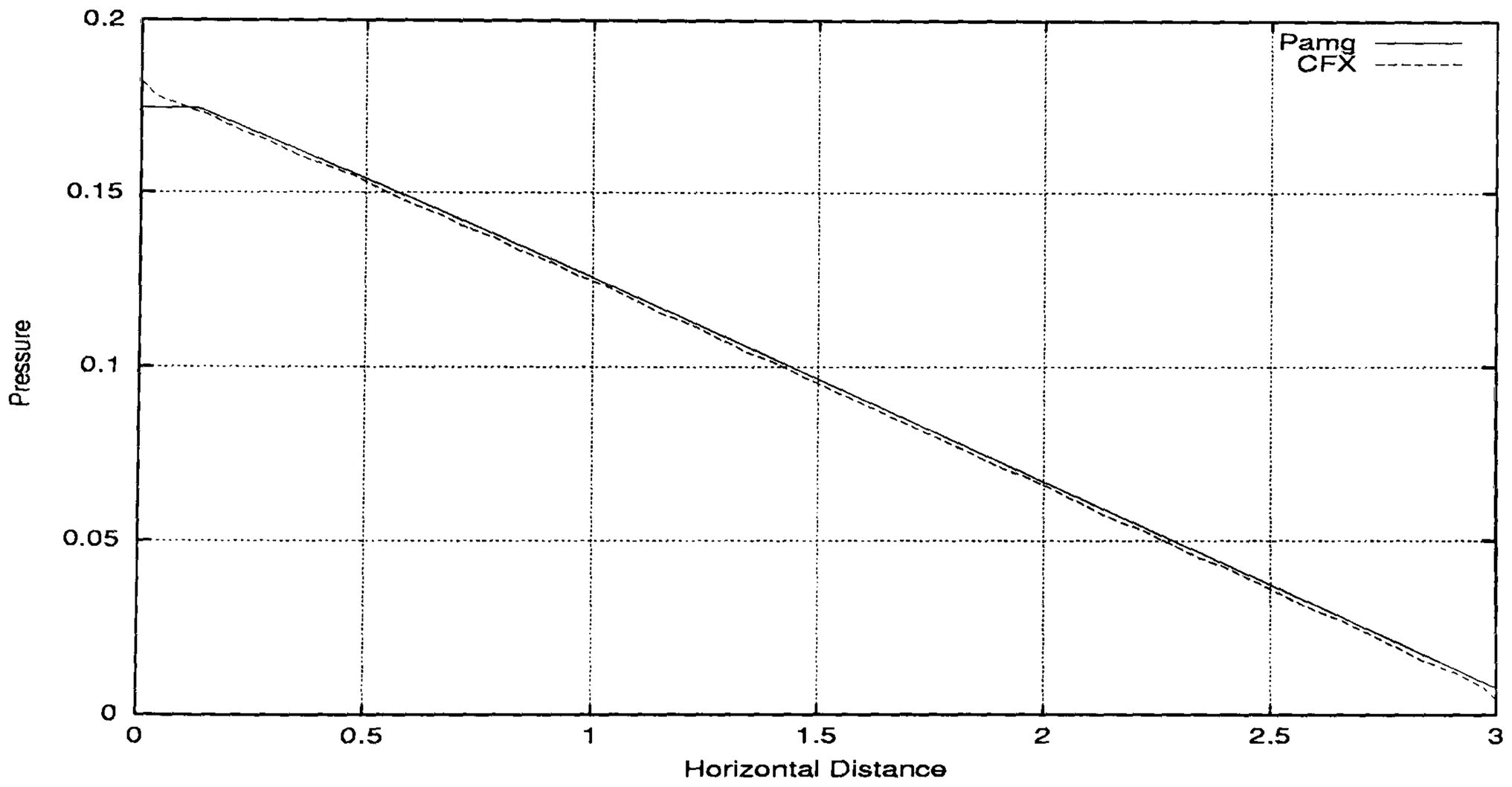


Figure 5.99: Multi-phase channel flow - pressure drop profile along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen the good agreement between the pressure drop computed by both solvers.

in a very positive way. However with respect to the time step independence the conclusion can not be so obvious. Evidence suggests that the volume fractions are less time step independent than the velocities. However as the time goes up the disagreement observed diminishes. This is what is expected since we are simulating steady flows. To measure correctly time step accuracy unsteady flows should be considered. However this kind of flows are outside the scope of this thesis.

5.3.2 Two-Phase Flow Through a T-Junction with Two Inlets

Problem definition

The second test problem to validate the `transient Pamg Multi-phase` is the simulation of two fluids trough a T-junction with two inlets. This is a much more difficult problem than the multi-phase channel flow. The existence of two inlets in this geometry originates that we have a region where the flux of the flow coming from one inlet joins the flux coming from the other inlet. Consequently the variations on the velocities and volume fractions can be very quickly and in the two directions of the axes. The physical properties of the two fluids are:

Two-Phase Flow	Reynolds Number	Viscosity	Density
Phase I	100	0.01	1.0
Phase II	75	0.0066	0.5

In this test case, the values chosen for the drag coefficient and the inter-facial length are:

$$C_D = 1.0$$

$$d_{\alpha\beta} = d_{\beta\alpha} = 0.1$$

Figure 5.100 shows the geometry of this test case. In Figure 5.101 we see the computational grid at the uniform level 3. The resolution of the coarsest grid is $\Delta_{xsize} = \Delta_{ysize} = 0.25$. The sections where the solution profiles are going to be presented are shown in Figure 5.102. The boundary conditions for the velocity are:

- In Inlet1 the horizontal velocity profile is defined for each phase as $u(0, y) = 1$ and the vertical velocity profile as $v(0, y) = 0$
- In Inlet2 the horizontal velocity profile is defined for each phase as $u(x, -2h) = 0$ and the vertical velocity profile as $v(x, -2h) = 1$

The volume fractions are set to 0.5 for each phase and for both the inlets. The initial condition is similar to the channel two-phase test case. The velocities are all set to zero and the volume fractions to 0.5. The convergence tolerance for the simulation is 10^{-6} for each time step.

Results for a 3 level computation

In Figures 5.103 and 5.104 we see the streamlines for the two phases after 20 time units. It can be observed that both phases of the fluid flow coming from the two different inlets join after the junction and consequently the streamlines go up in the y direction.

Figures 5.105 through 5.111 present the profiles of the horizontal velocity, vertical velocity, volume fraction and pressure drop along the lines $x = 3.5$ and $y = 0.5$

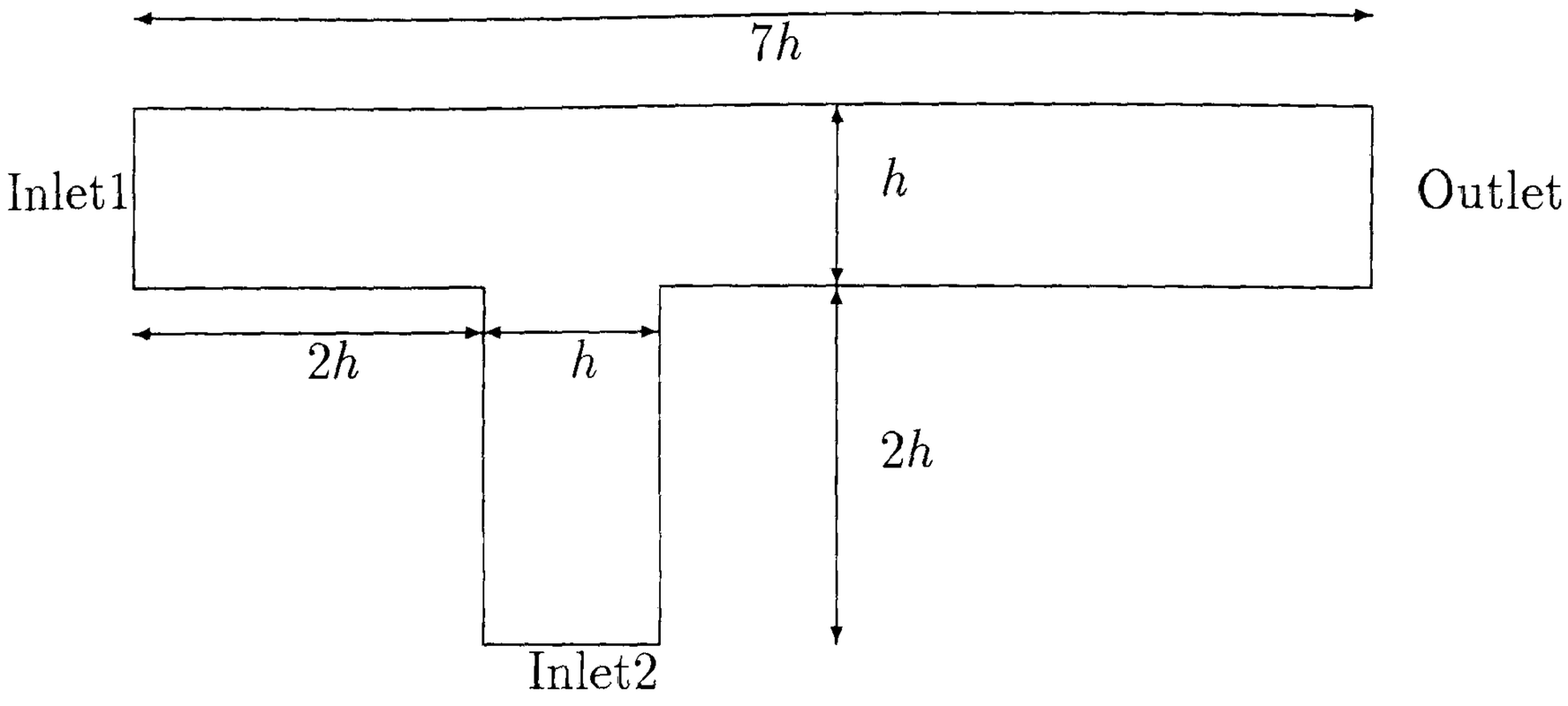


Figure 5.100: Geometrical representation of the domain of the T-junction flow problem

for 3 level computations. In Figure 5.105 we can see that the horizontal velocity increases substantially between $x = 2.0$ and $x = 3.0$ which is precisely where the fluid flow coming from the inlet2 joins the fluid coming from inlet1. At the outlet we see the separation of the two phases even at $t = 1$ time unit.

Figure 5.106 presents the vertical velocity profile along the line $y = 0.5$ for 3 level computation. It is visible the increase in the vertical velocity after $x = 2.0$. After $x = 3.0$ the vertical velocity goes down and even takes negative values for both phases.

In Figure 5.107 the volume fractions along the line $y = 0.5$ varies substantially after the point $x = 2.5$ which is precisely where the two flows coming from the two inlets are joining.

Figure 5.108 presents the horizontal velocity along the line $x = 3.5$. In the walls it can be seen that the velocity at $y = 0.0$ is slightly smaller than at $y = 1.0$. Hence the velocity profile is not totally symmetric.

In Figure 5.109 it can be seen that the vertical velocity of *Phase II* at 20 time units is substantially bigger than the corresponding vertical velocity of *Phase I* at 20 time units.

In Figure 5.110 we present the volume fraction profile along the line $x = 3.5$ for 3 level computation. It is visible that from 1 to 20 time units the difference between the volume fractions in the wall $y = 0.0$ becomes bigger. At this wall the *Phase II* fills almost all the volume fraction.

Figure 5.111 presents the pressure drop profile along the line $y = 0.5$ for 3 level computation. It can be seen the slight slope between 2.5 and 3.5 which corresponds to the region where the two fluids join. Naturally in the other regions of the domain the pressure drop varies linearly and consequently the corresponding slopes are straight lines.

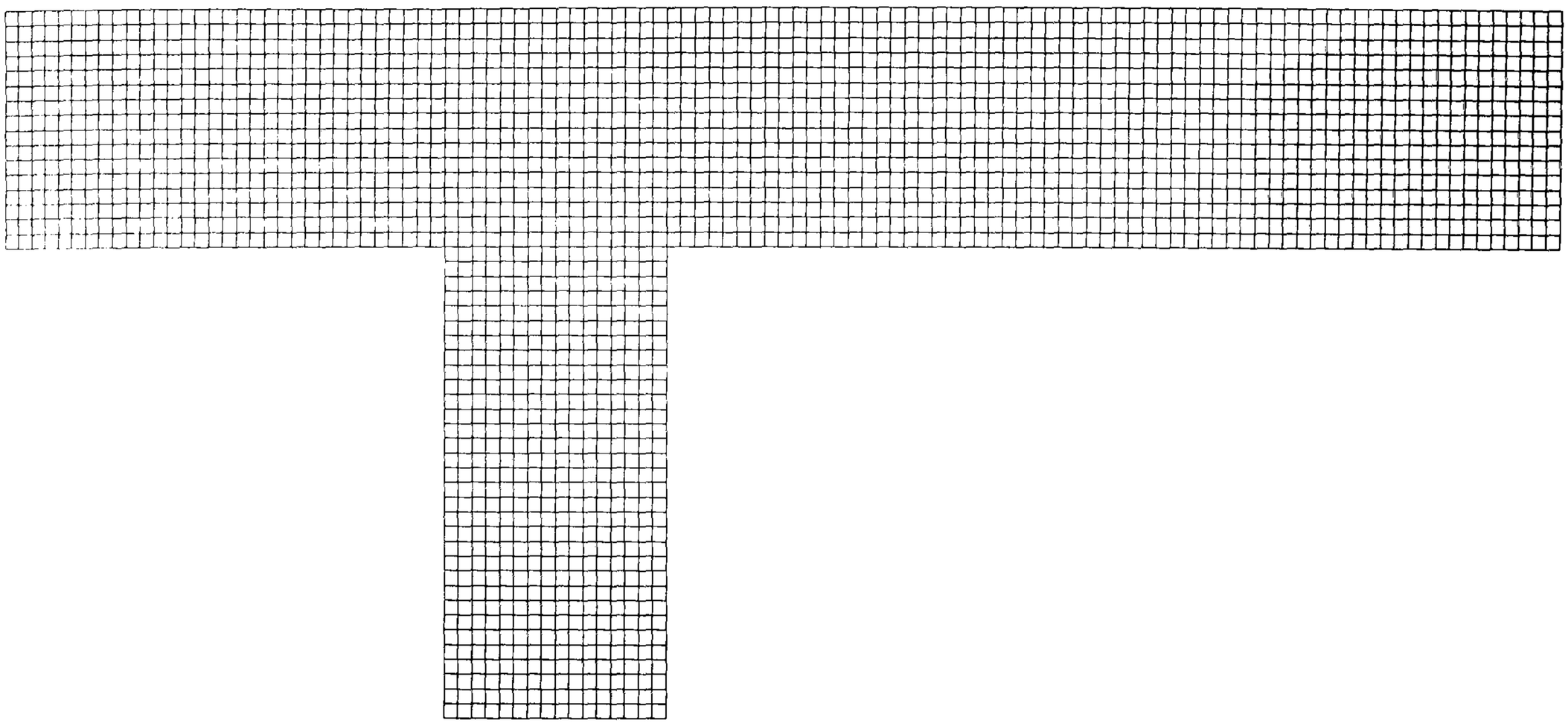


Figure 5.101: Uniform level 3 grid for the T-Junction flow test case ($\Delta_{xsize} = 0.0625$ and $\Delta_{ysize} = 0.0625$)

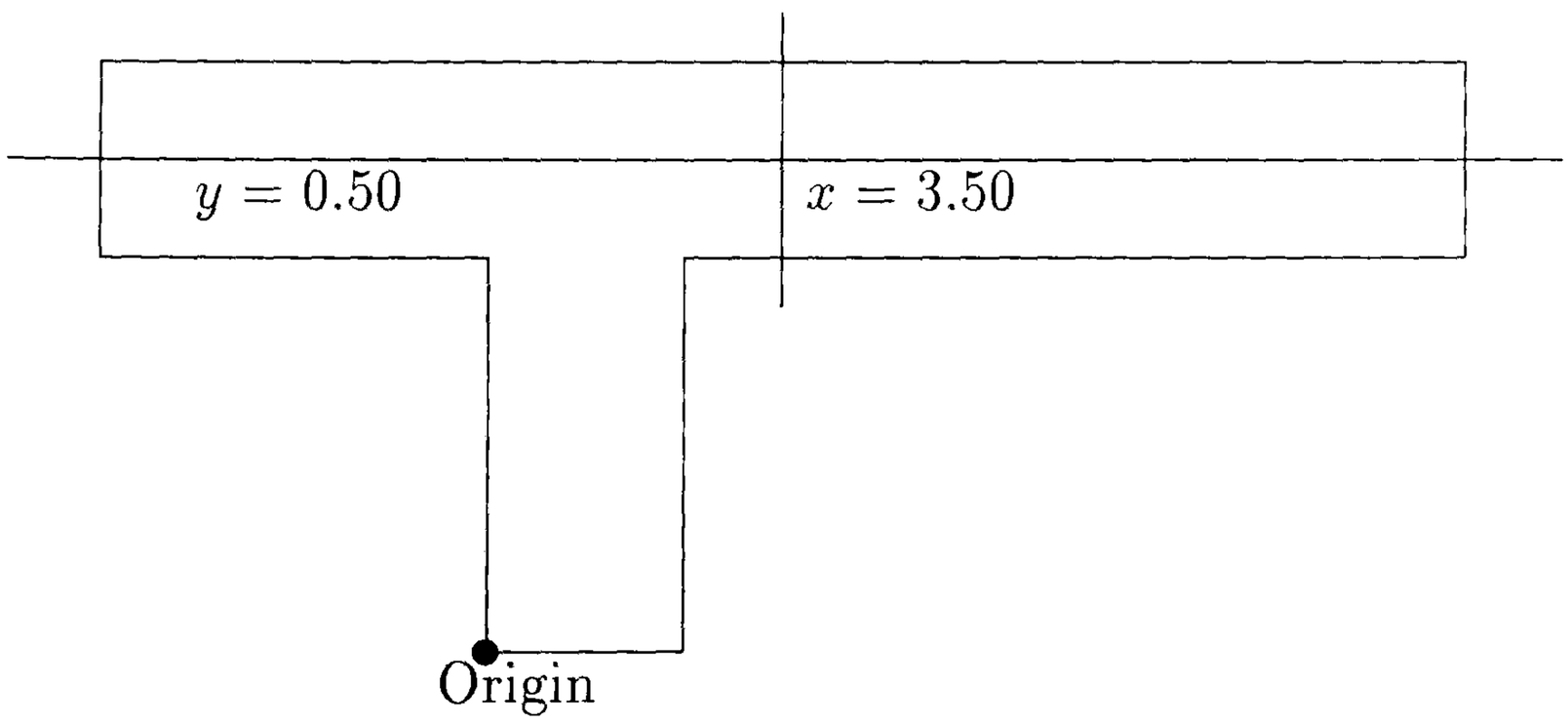


Figure 5.102: T-junction Flow test case - main sections used to give the solution profiles

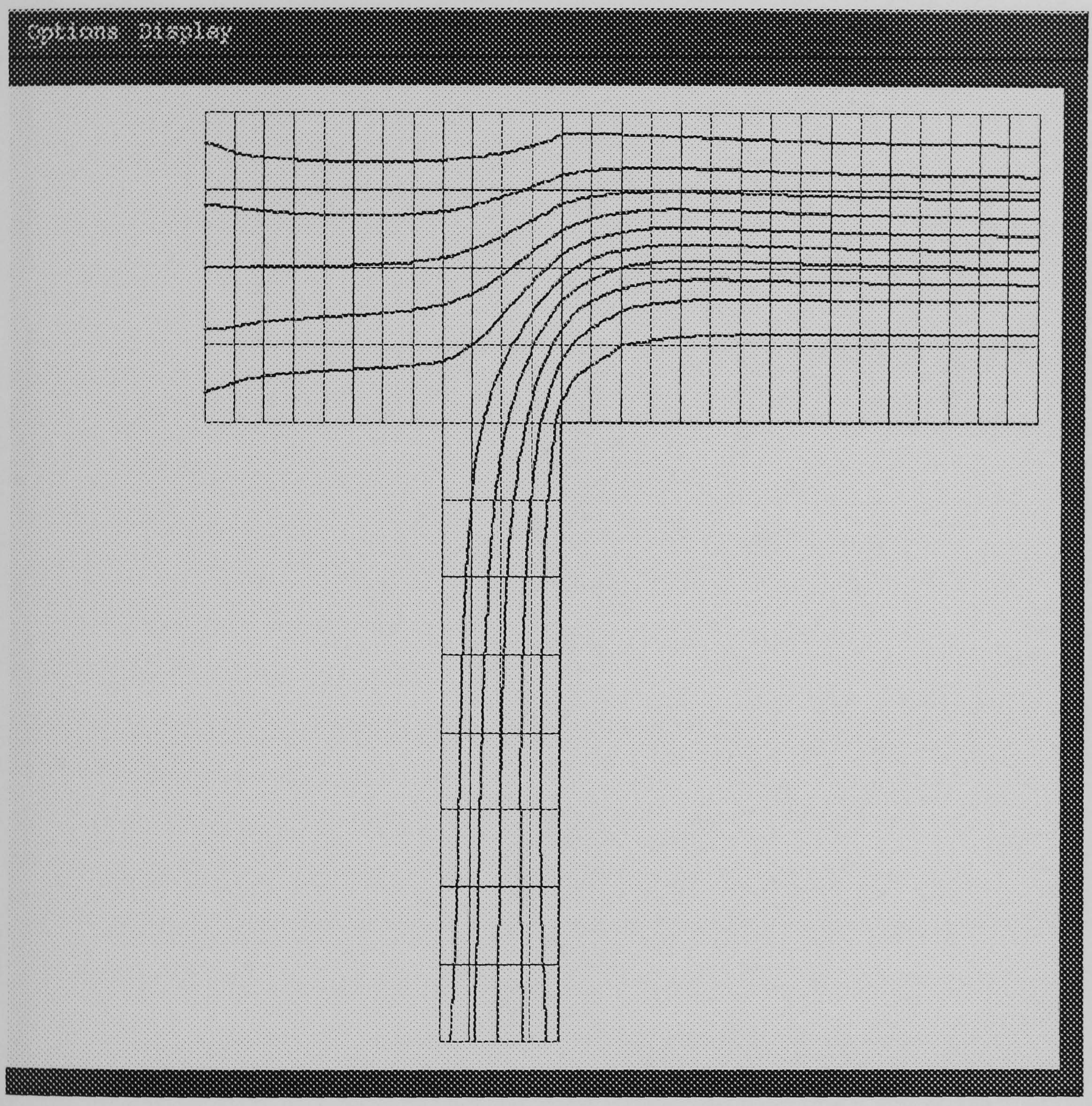


Figure 5.103: Multi-phase T-junction flow - streamlines after 20 time units for *Phase I*

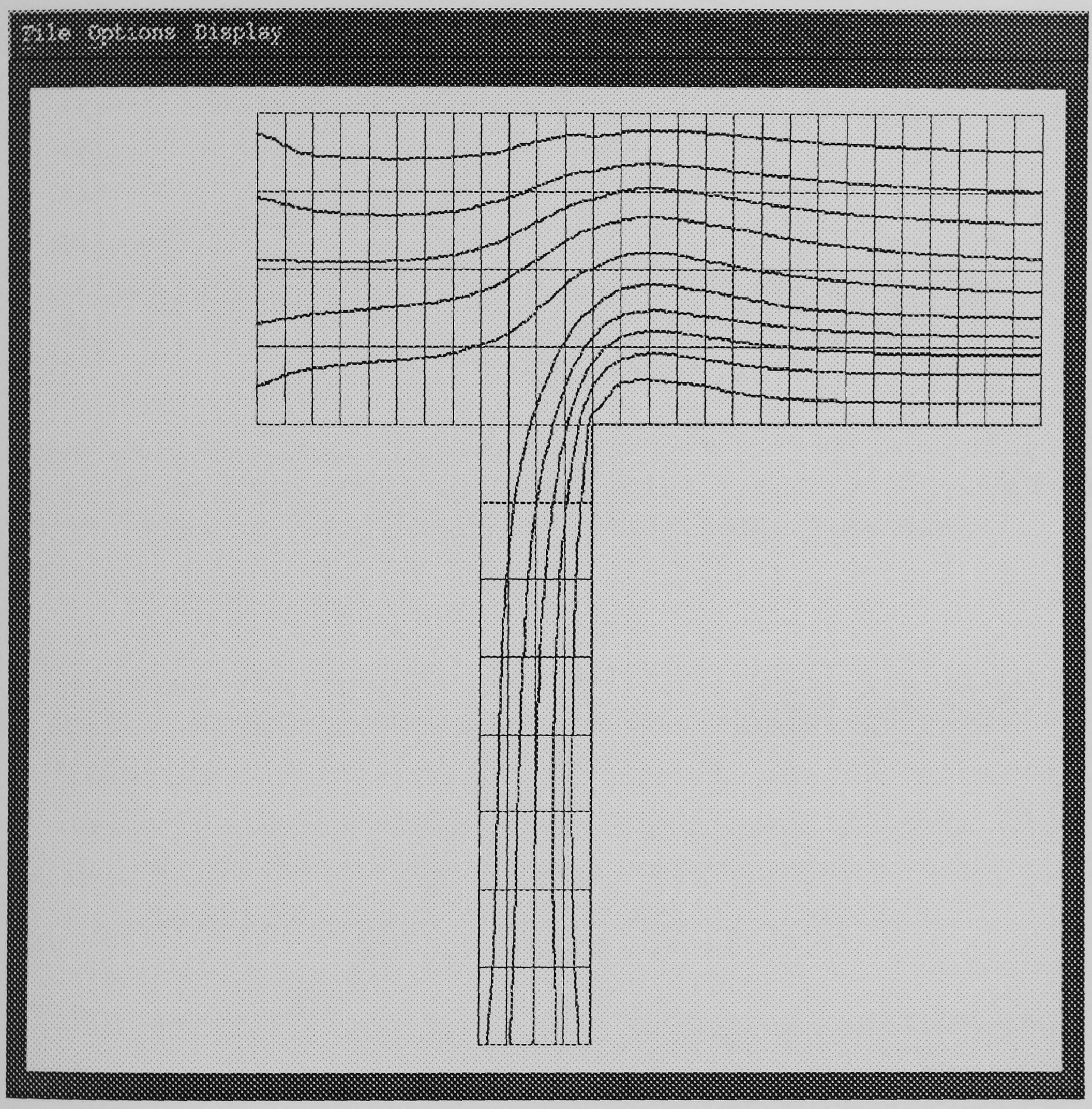


Figure 5.104: Multi-phase T-junction flow - streamlines after 20 time units for *Phase II*

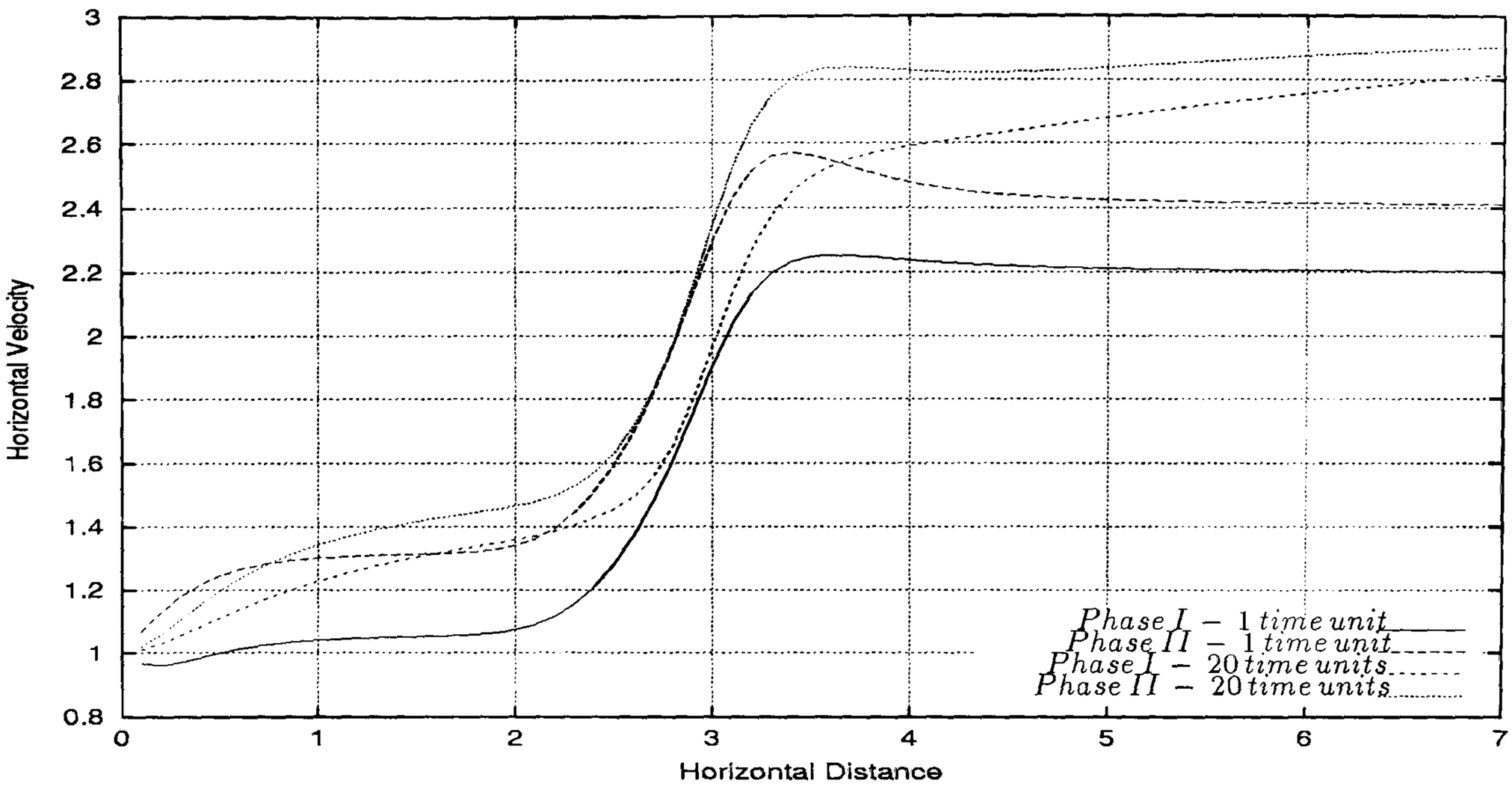


Figure 5.105: Multi-phase T-junction flow - horizontal velocity profile along the line $y = 0.5$ for 3 level computation. It can be seen the increase of the velocities between 2.0 and 3.0. Note the separation of the two phases at the outlet.

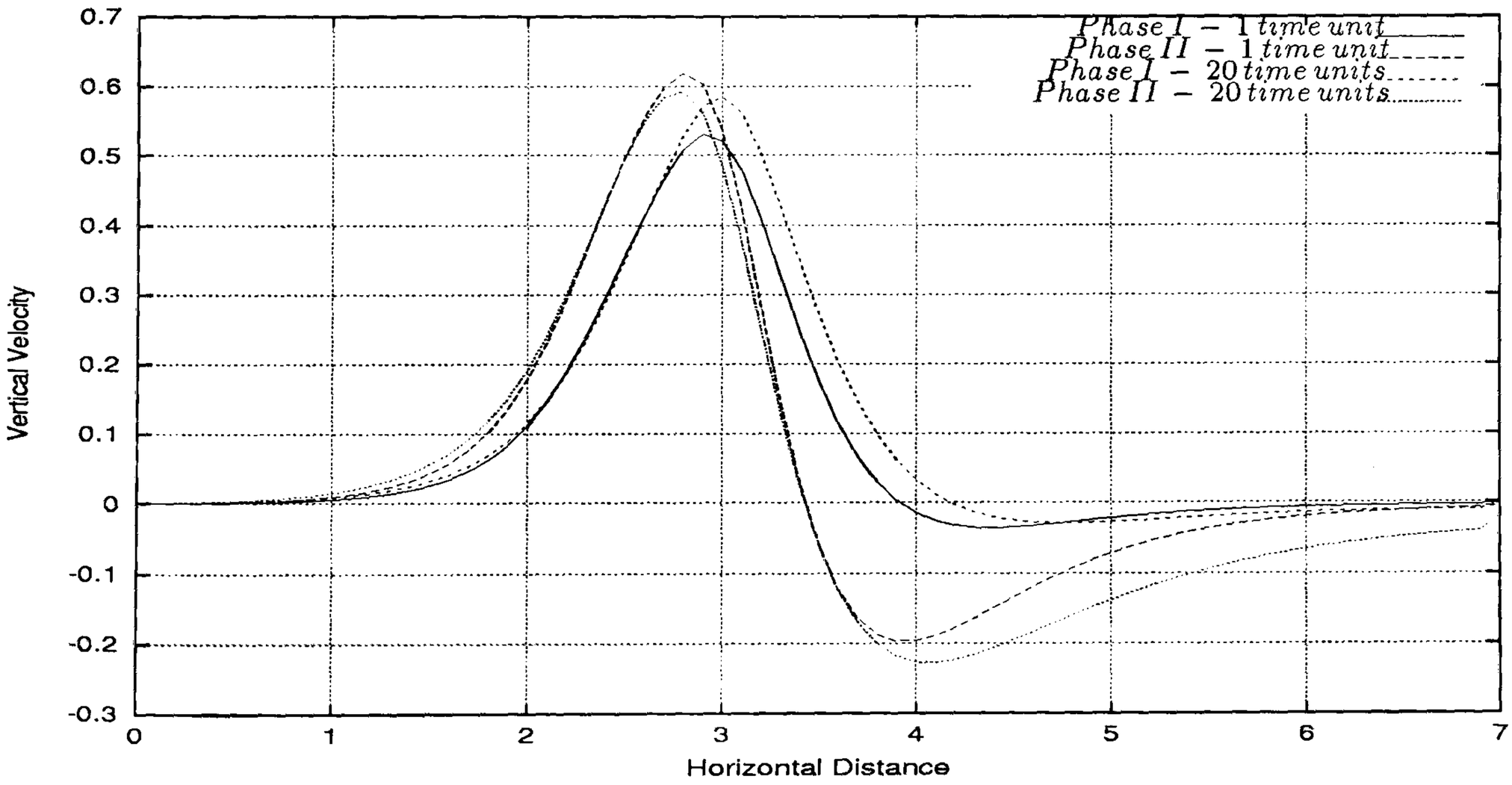


Figure 5.106: Multi-phase T-junction flow - vertical velocity profile along the line $y = 0.5$ for 3 level computation. It can be seen the increase in the vertical velocity after $x = 2.0$. After 3.0 it can be observed the decrease to negative values by both phases.

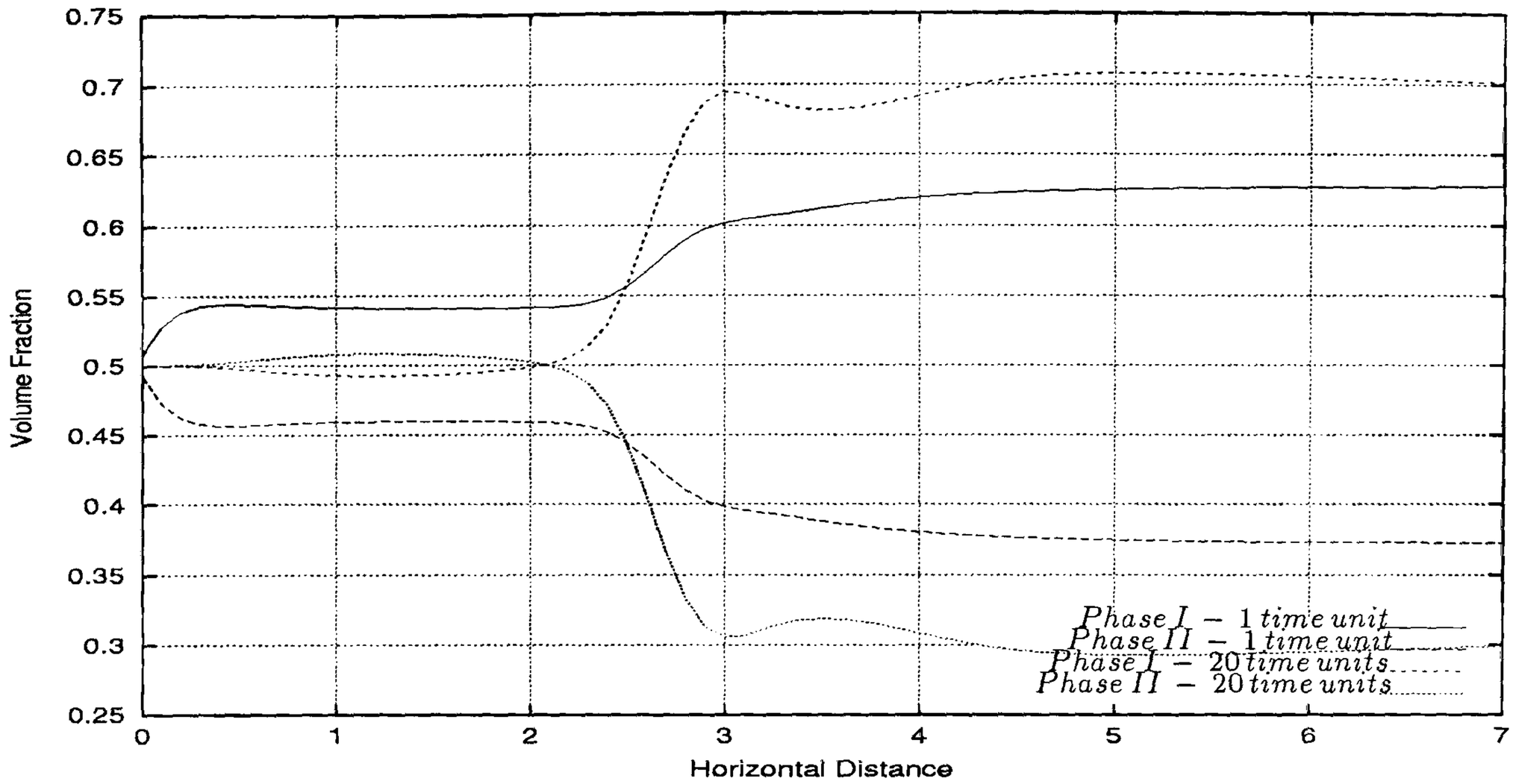


Figure 5.107: Multi-phase T-junction flow - volume fraction profile along the line $y = 0.5$ for 3 level computation. It can be observed the big variation of the volume fractions after $x = 2.5$.

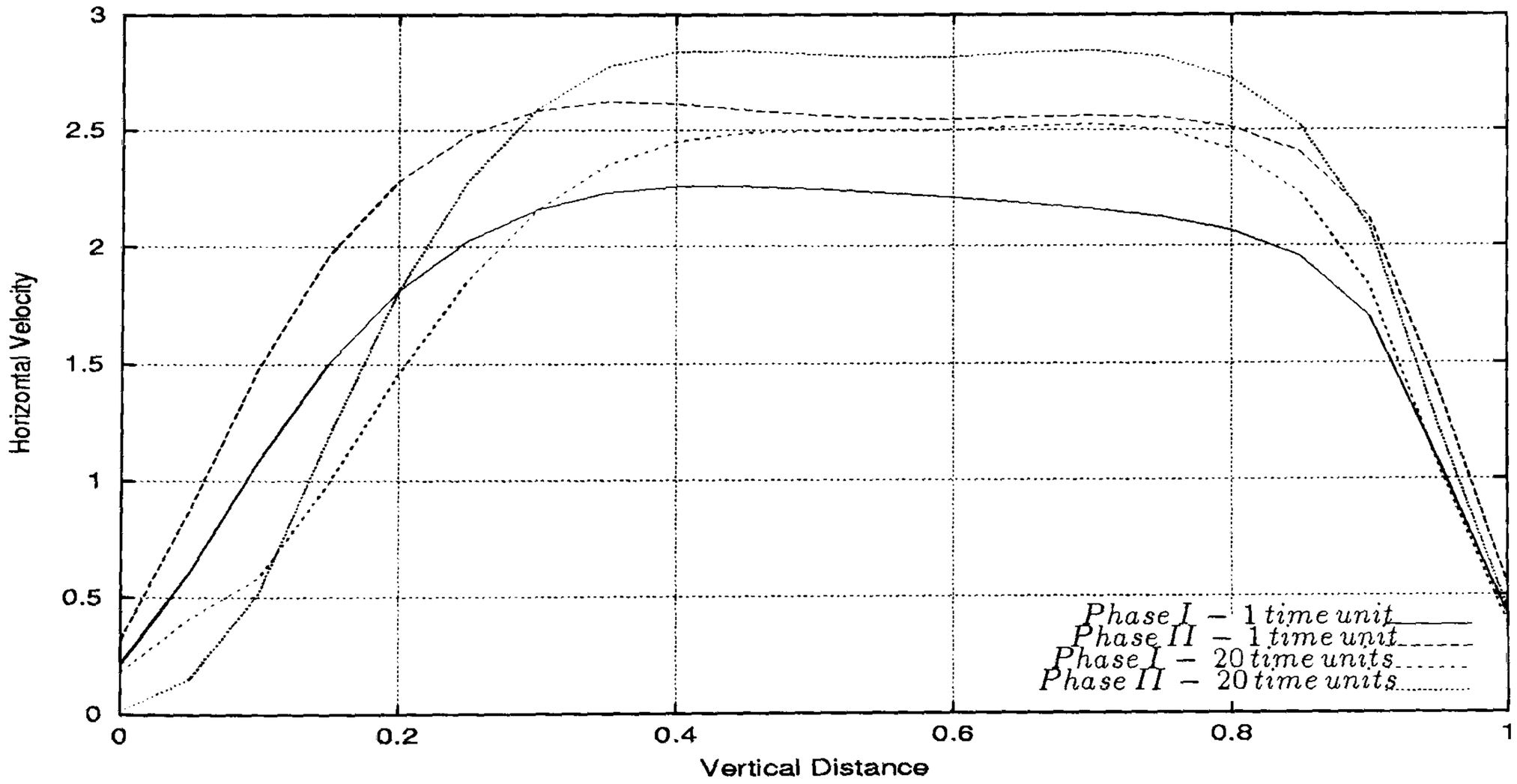


Figure 5.108: Multi-phase T-junction flow - horizontal velocity profile along the line $x = 3.5$ for 3 level computation. It can be seen that the velocity profile is not totally symmetric.

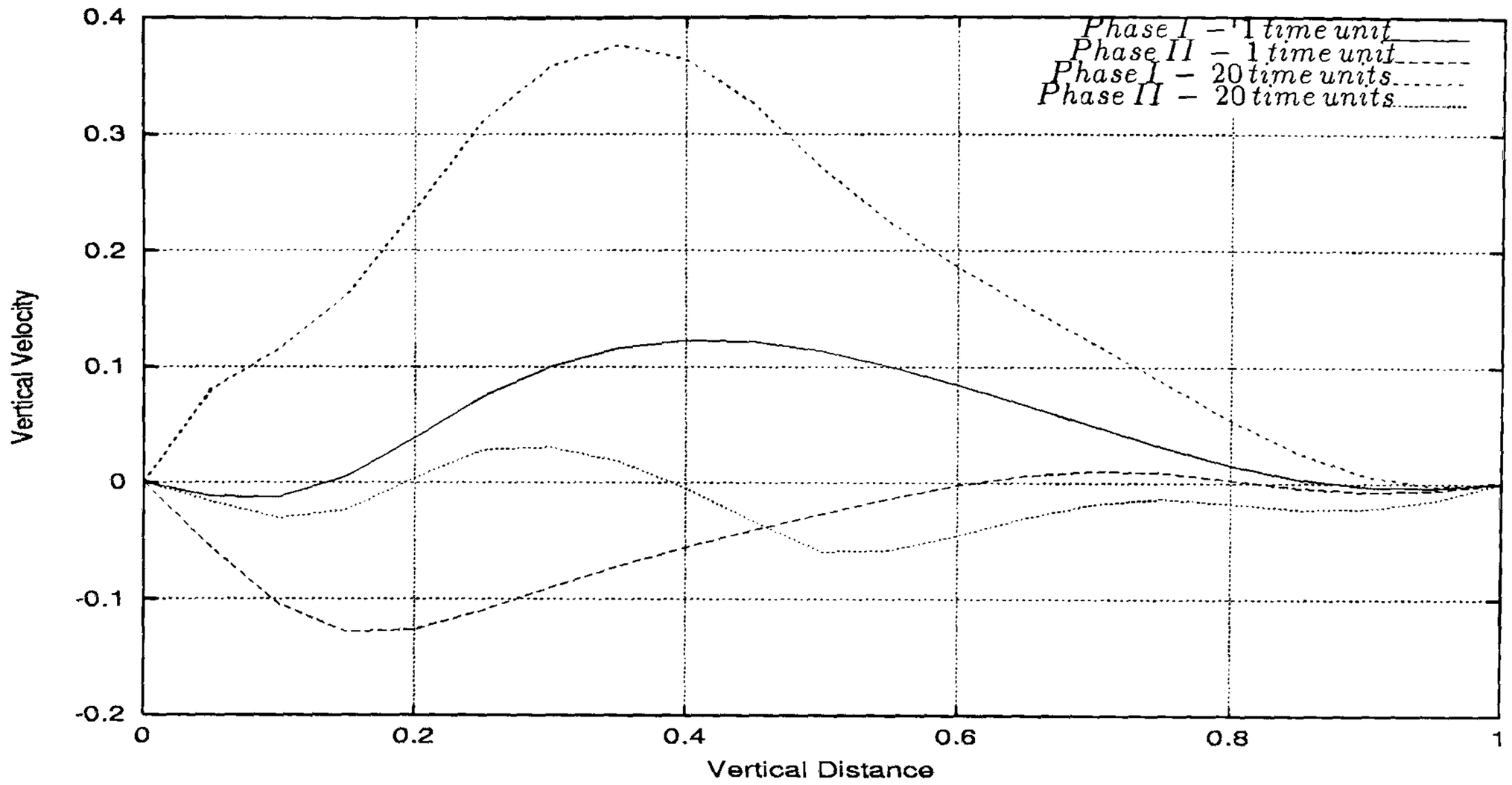


Figure 5.109: Multi-phase T-junction flow - vertical velocity profile along the line $x = 3.5$ for 3 level computation. It can be observed that the vertical velocity of *Phase II* is substantially bigger than the vertical velocity of *Phase I* at 20 time units.

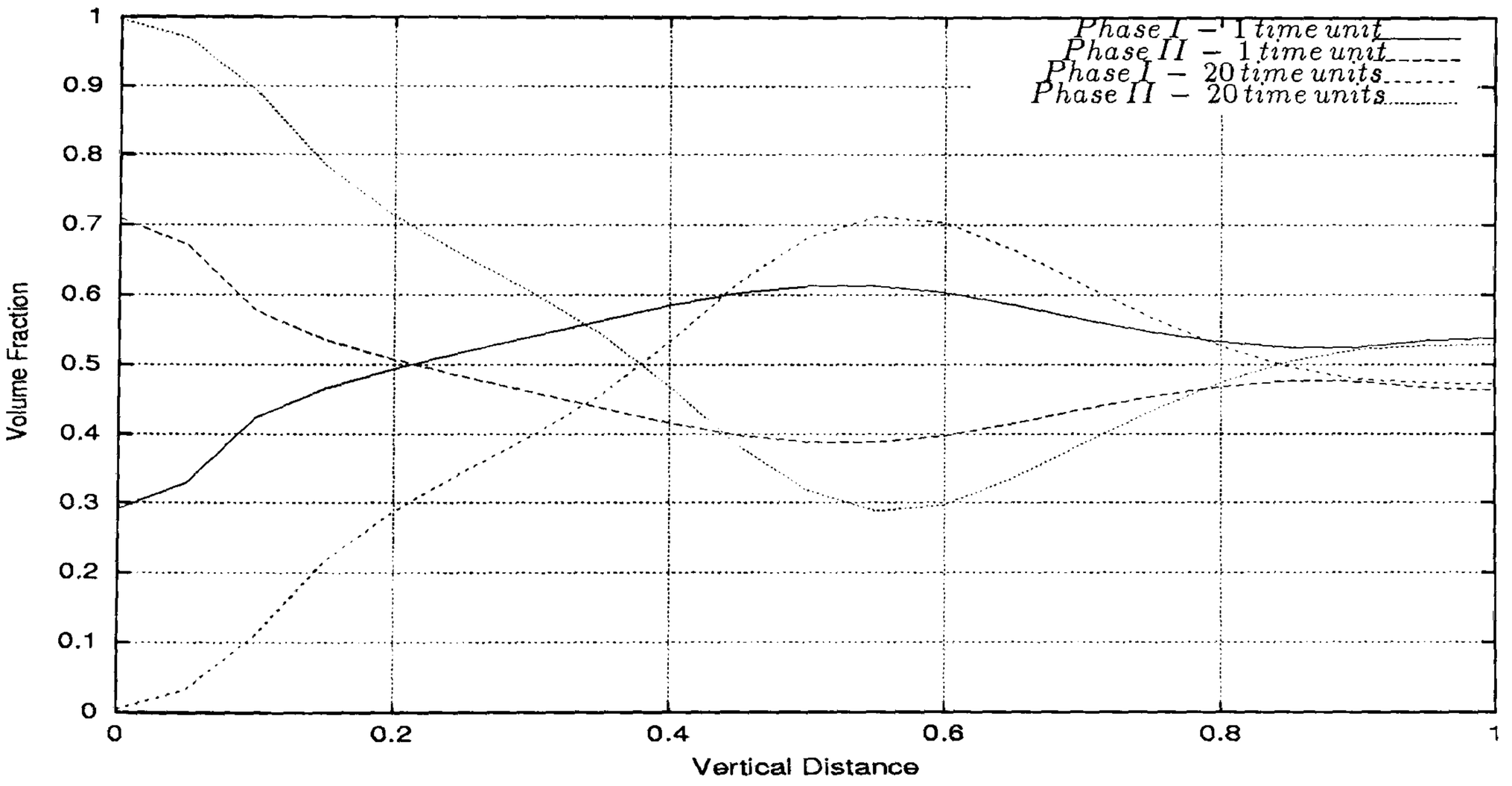


Figure 5.110: Multi-phase T-junction flow - volume fraction profile along the line $x = 3.5$ for 3 level computation. It can be seen that from 1 to 20 time units the difference between the volume fractions in the wall $y = 0.0$ becomes bigger.

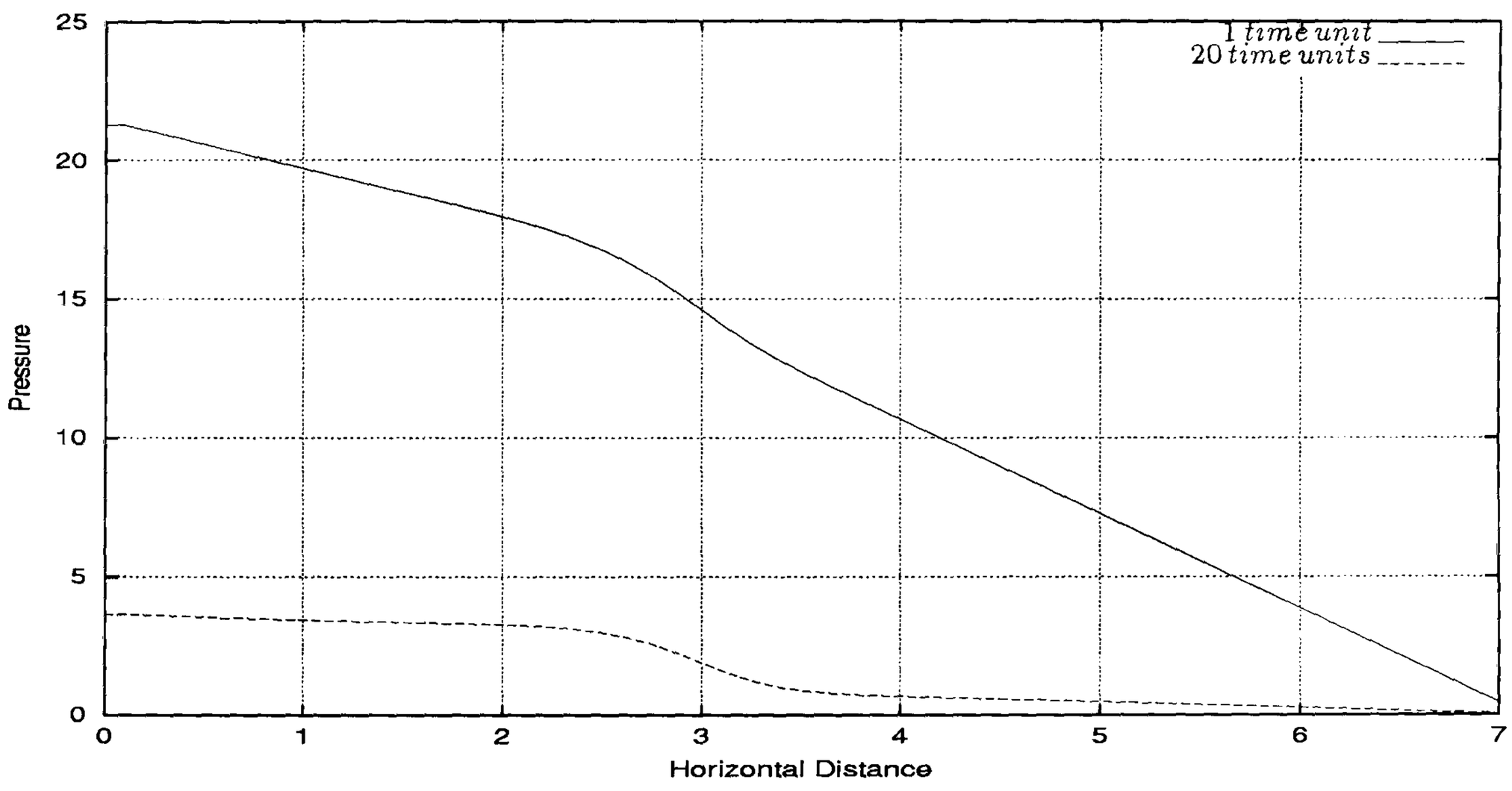


Figure 5.111: Multi-phase T-junction flow - pressure drop profile along the line $y = 0.5$ for 3 level computation. It can be observed the slight slopes between 2.5 and 3.5 where the two fluid flows join.

Grid and Time Step Independence of the Solutions

Figures 5.112 through 5.124 present the profiles of the horizontal velocity, vertical velocity, volume fraction and pressure drop along the lines $x = 3.5$ and $y = 0.5$ for single 2 and 3 level computations.

In Figure 5.112 it is visible that after 20 time units the curves corresponding to single 2 and 3 level computations change slightly due to the increasing values of the horizontal velocity. However the same sequence of the growing velocities observed in the single 2 and 3 level computations after 1 time unit is observed. In Figure 5.113 identical behaviour of the horizontal velocity for *Phase II* is observed.

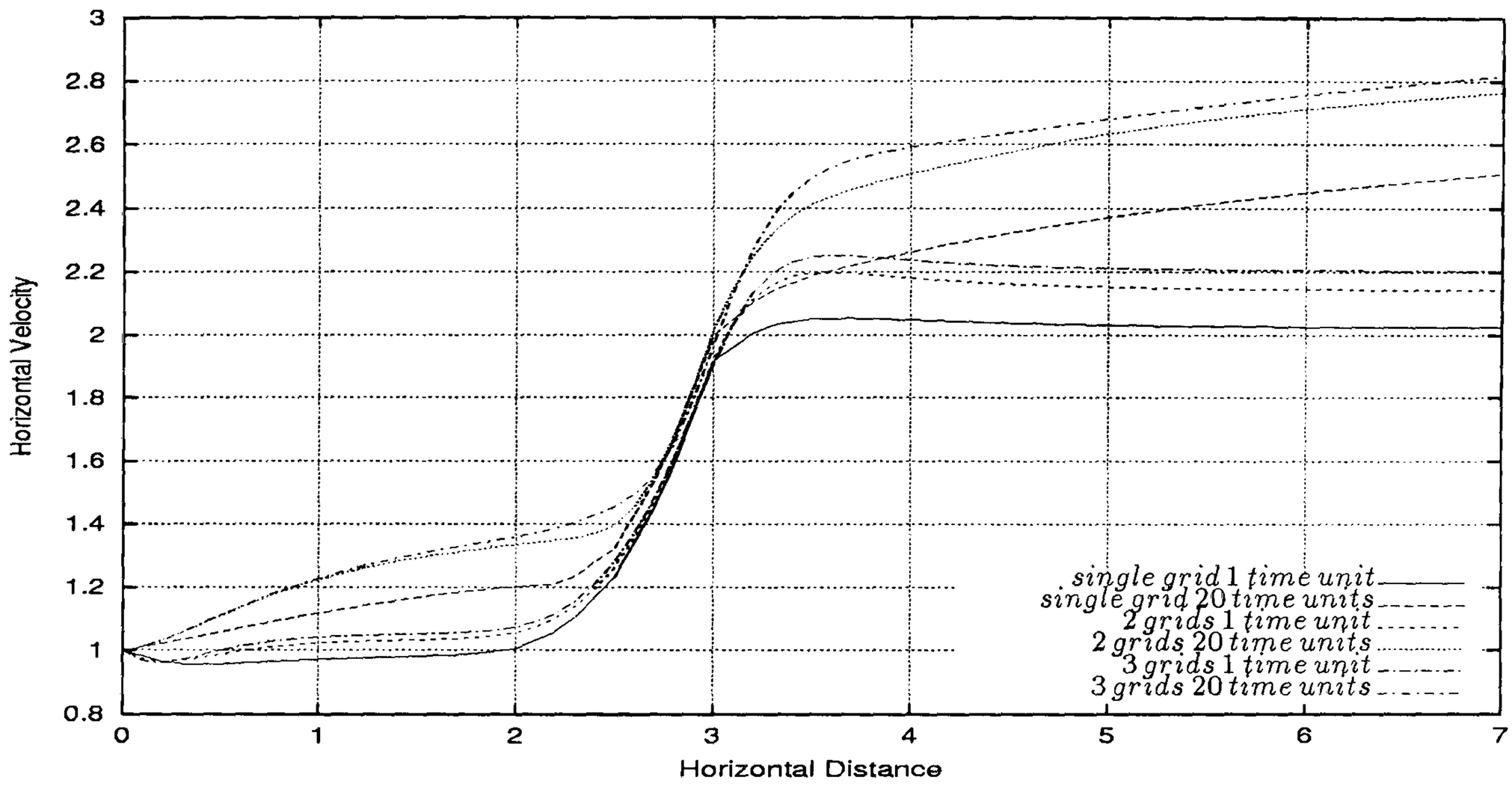


Figure 5.112: Multi-phase T-junction flow - comparison of horizontal velocity profiles for *Phase I* along the line $y = 0.5$ for single 2 and 3 level computations. It can be observed the same sequence of growing velocities for 1 and 20 time units for all level calculations.

The variation of the volume fractions along the line $y = 0.5$ (see Figures 5.114 and 5.115) is independent of the number of levels used in the computation. The volume fraction of *Phase I* goes up when the horizontal distance increases while the volume fraction of *Phase II* decreases. This behaviour is observed for all the grid levels and for 1 and 20 time units. Comparing the variations of the horizontal velocities and volume fractions in the outlet for single 2 and 3 level computations two different kinds of behaviour can be observed. The horizontal velocities for 2 and 3 grids are very close at the outlet while the volume fractions are not so close. This seems to suggest that the horizontal velocities are second order accurate while the volume fractions are first order.

In Figures 5.116 and 5.117 we present the horizontal velocity profiles for *Phase I* and *Phase II* along the line $x = 3.375$ for single 2 and 3 level computations. The parabolic profile that can be seen is independent of the number of grids at 1 and 20 time units.

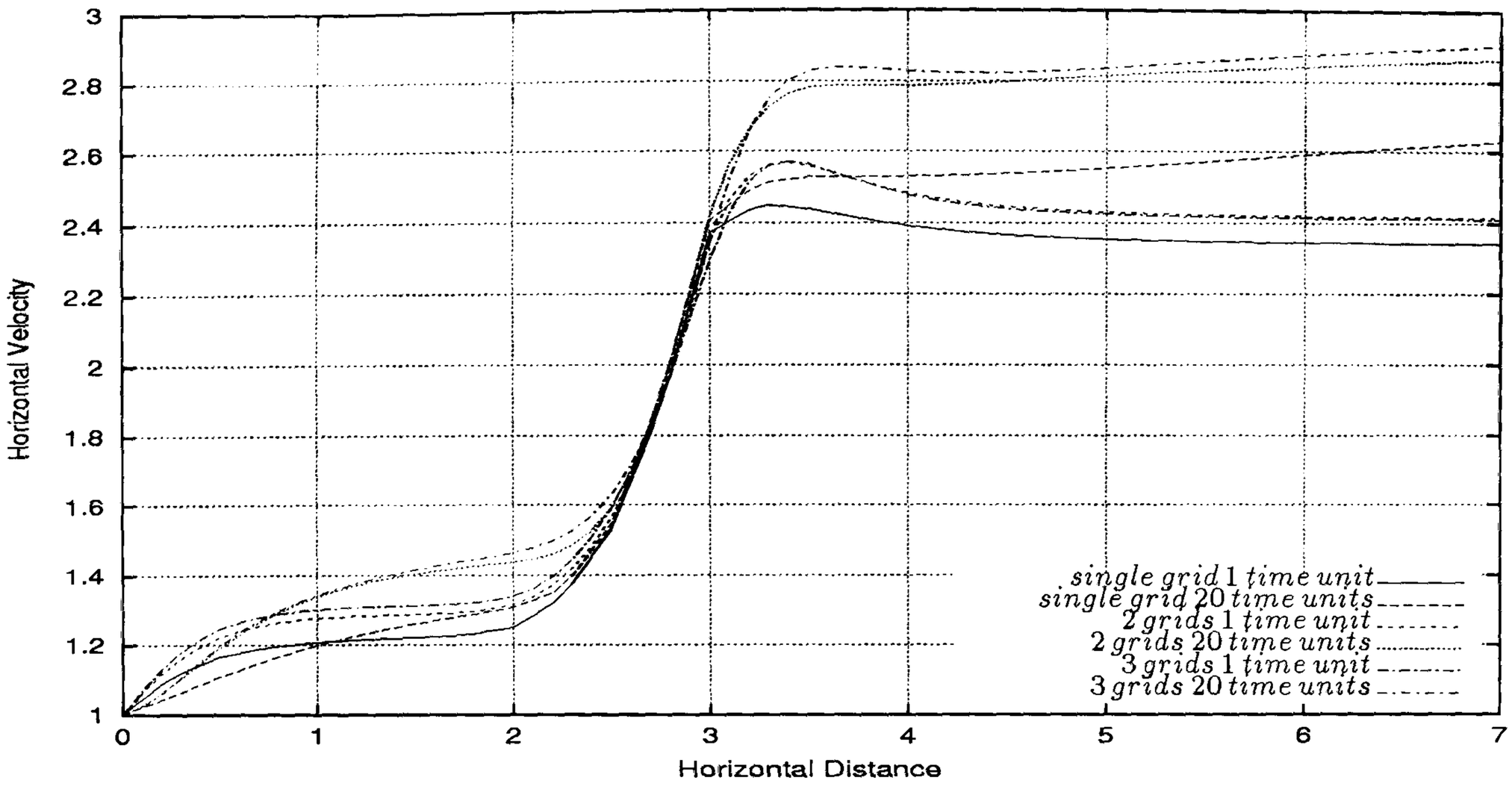


Figure 5.113: Multi-phase T-junction flow - comparison of horizontal velocity profiles for *Phase II* along the line $y = 0.5$ for single 2 and 3 level computations. It can be seen the same sequence of growing velocities for 1 and 20 time units for all level calculations.

Figures 5.118 and 5.119 show the vertical velocity profiles along the line $x = 3.375$ for single 2 and 3 level computations. The curves for single 2 and 3 level computations at 20 time units exhibit the same pattern as at 1 time unit. Consequently the grid independence of the solution algorithm is verified.

In Figures 5.120 and 5.121 we present the volume fraction profiles for *Phase I* and *Phase II* along the line $x = 3.375$ for single 2 and 3 level computations. It can be observed that the curves for both phases at 20 and 1 time units have an equivalent pattern over all the entire vertical distance.

Figures 5.122 and 5.123 show the vertical velocity profiles for *Phase I* and *Phase II* along the line $y = 0.75$ for single 2 and 3 level computations. It is visible that the vertical velocity reach the maximum value close to the point $x = 2.8$ for single 2 and 3 grids computation. Moreover at the inlet and outlet it is null for every level computation.

The variation of the pressure (see Figure 5.124) decreases as the time increases from 1 to 20 time units. Identical behaviour is observed for both the single 2 and 3 level computations. Hence the grid independence of the solution algorithm can be established.

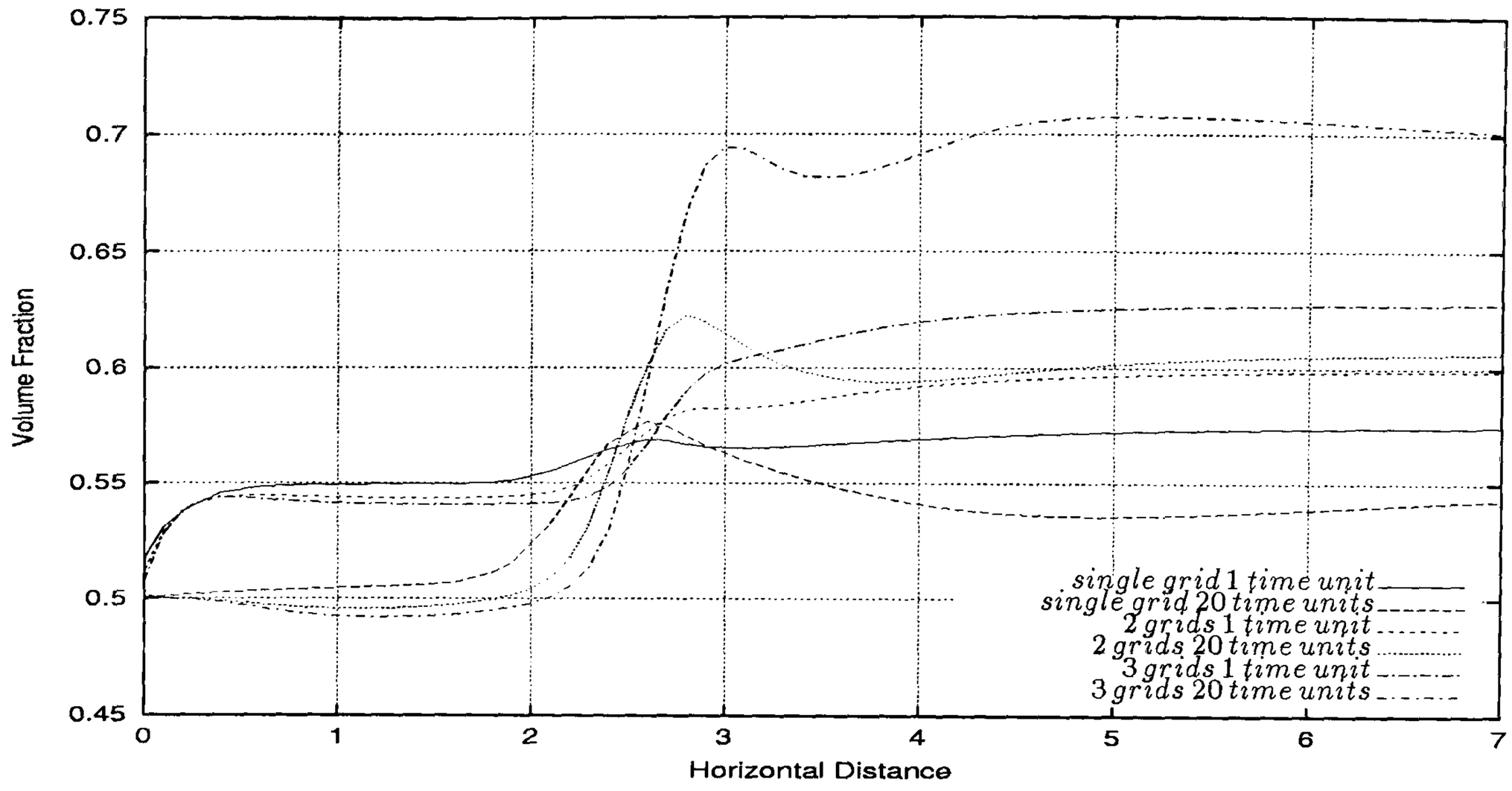


Figure 5.114: Multi-phase T-junction flow - comparison of volume fraction profiles for *Phase I* along the line $y = 0.5$ for single 2 and 3 level computations. It can be observed that the volume fraction goes up for all level calculations.

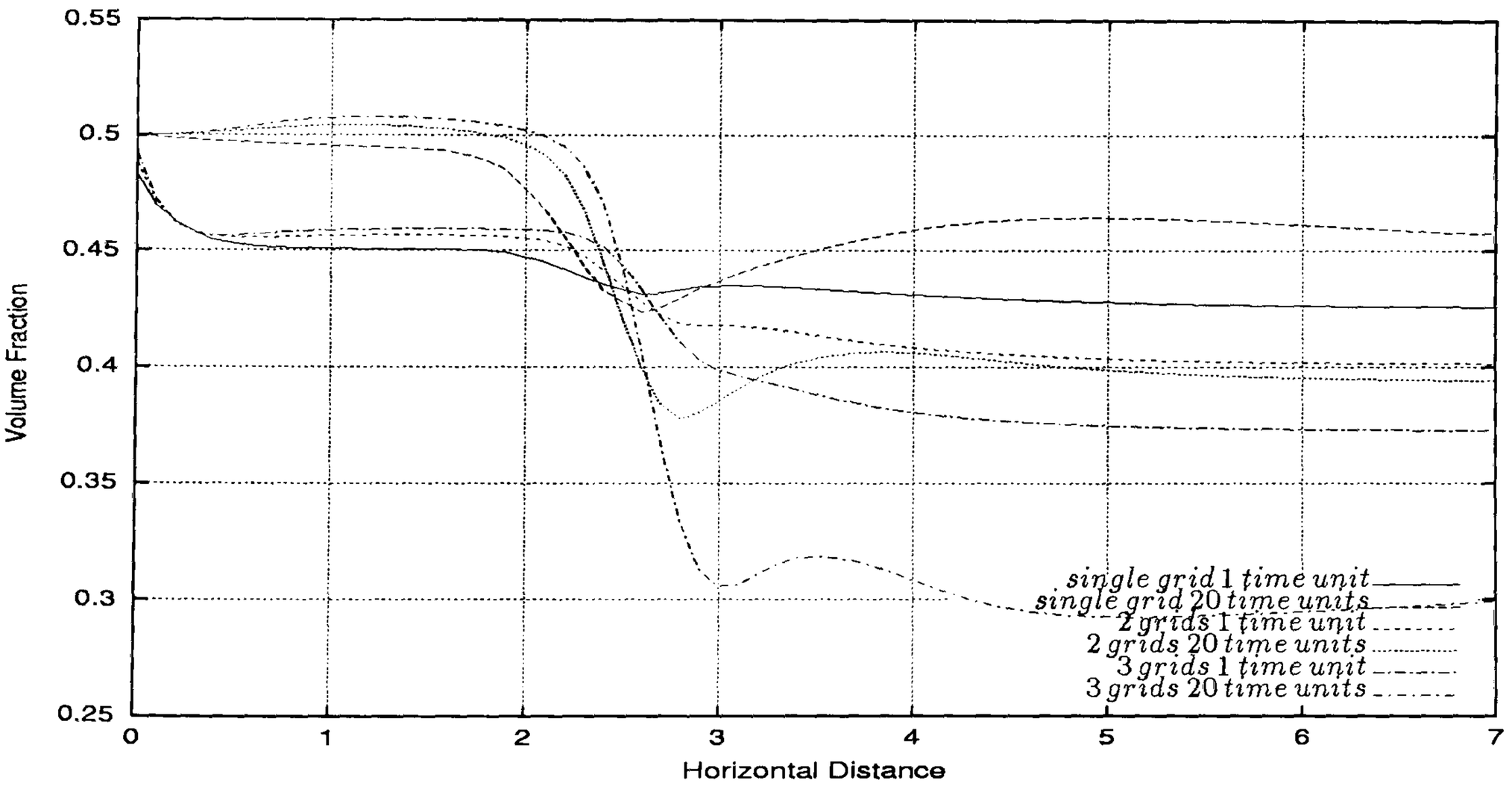


Figure 5.115: Multi-phase T-junction flow - comparison of volume fraction profiles for *Phase II* along the line $y = 0.5$ for single 2 and 3 level computations. It can be seen that the volume fraction goes down for all level calculations.

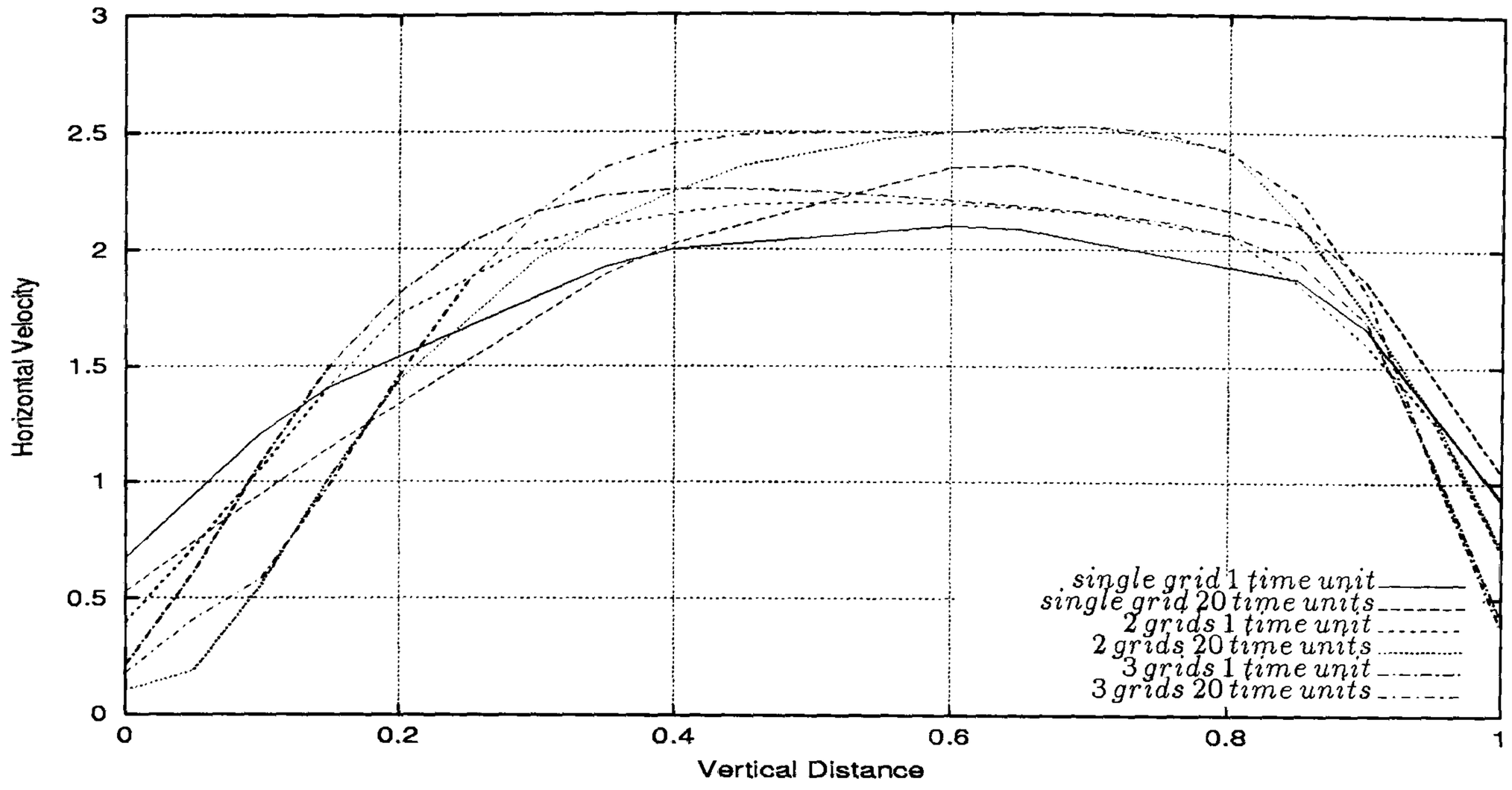


Figure 5.116: Multi-phase T-junction flow - comparison of horizontal velocity profiles for *Phase I* along the line $x = 3.375$ for single 2 and 3 level computations. It can be observed that the horizontal velocities exhibit parabolic profiles for all level calculations.

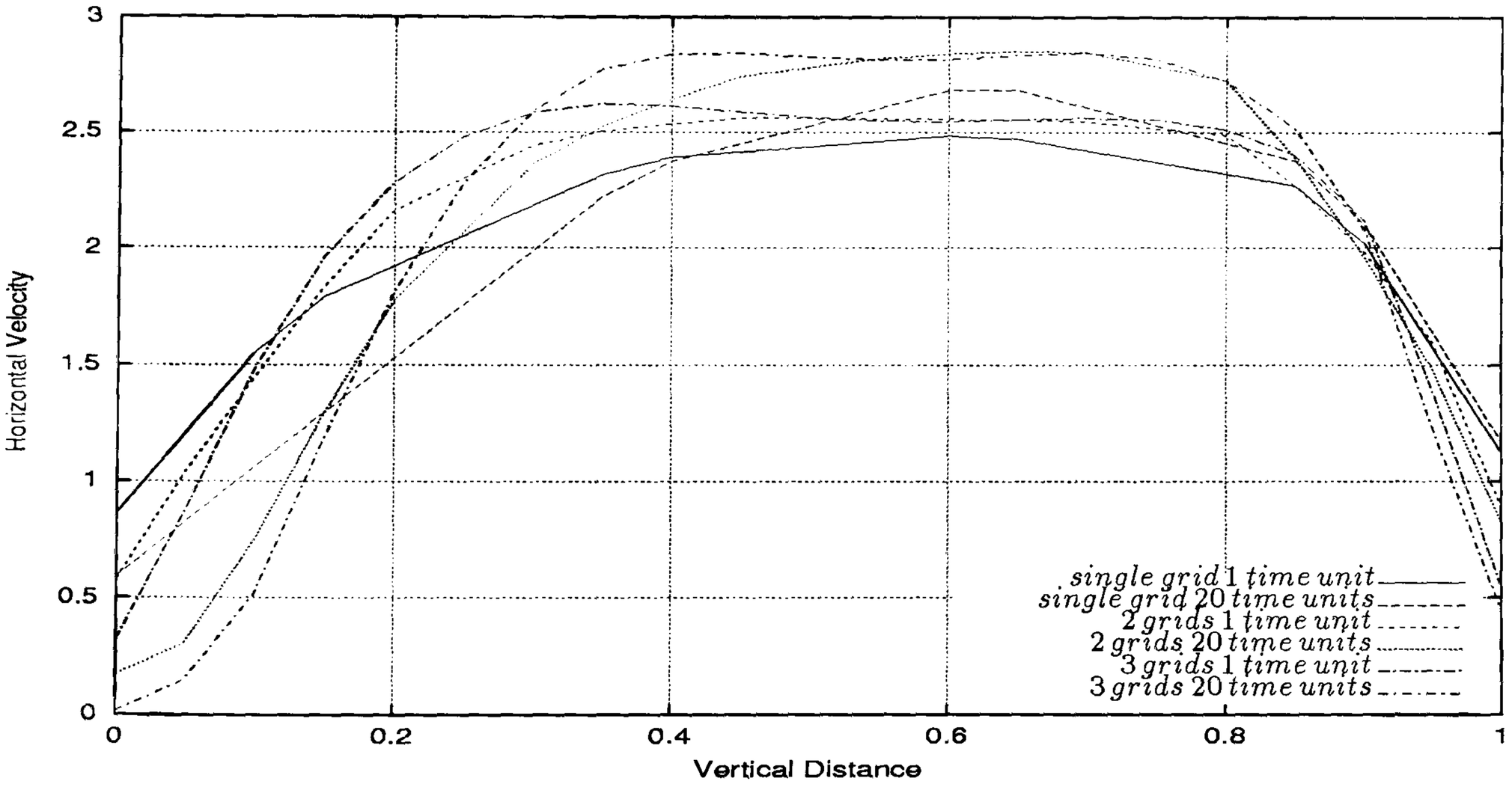


Figure 5.117: Multi-phase T-junction flow - comparison of horizontal velocity profiles for *Phase II* along the line $x = 3.375$ for single 2 and 3 level computations. It can be seen that the horizontal velocities exhibit parabolic profiles for all level calculations.

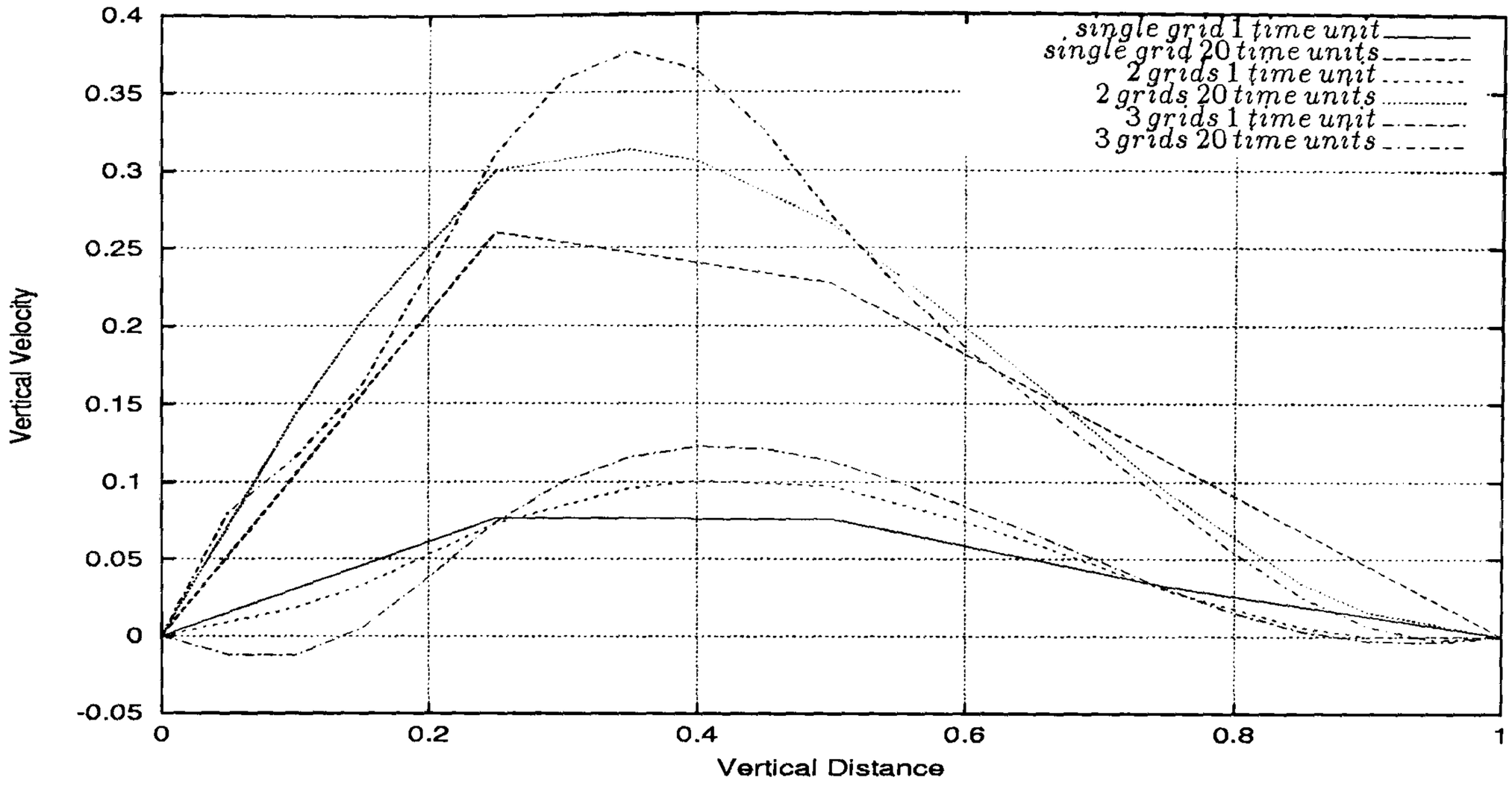


Figure 5.118: Multi-phase T-junction flow - comparison of vertical velocity profiles for *Phase I* along the line $x = 3.375$ for single 2 and 3 level computations. It can be seen that the curves for single 2 and 3 level computations at 20 time units exhibit the same pattern as at 1 time unit.

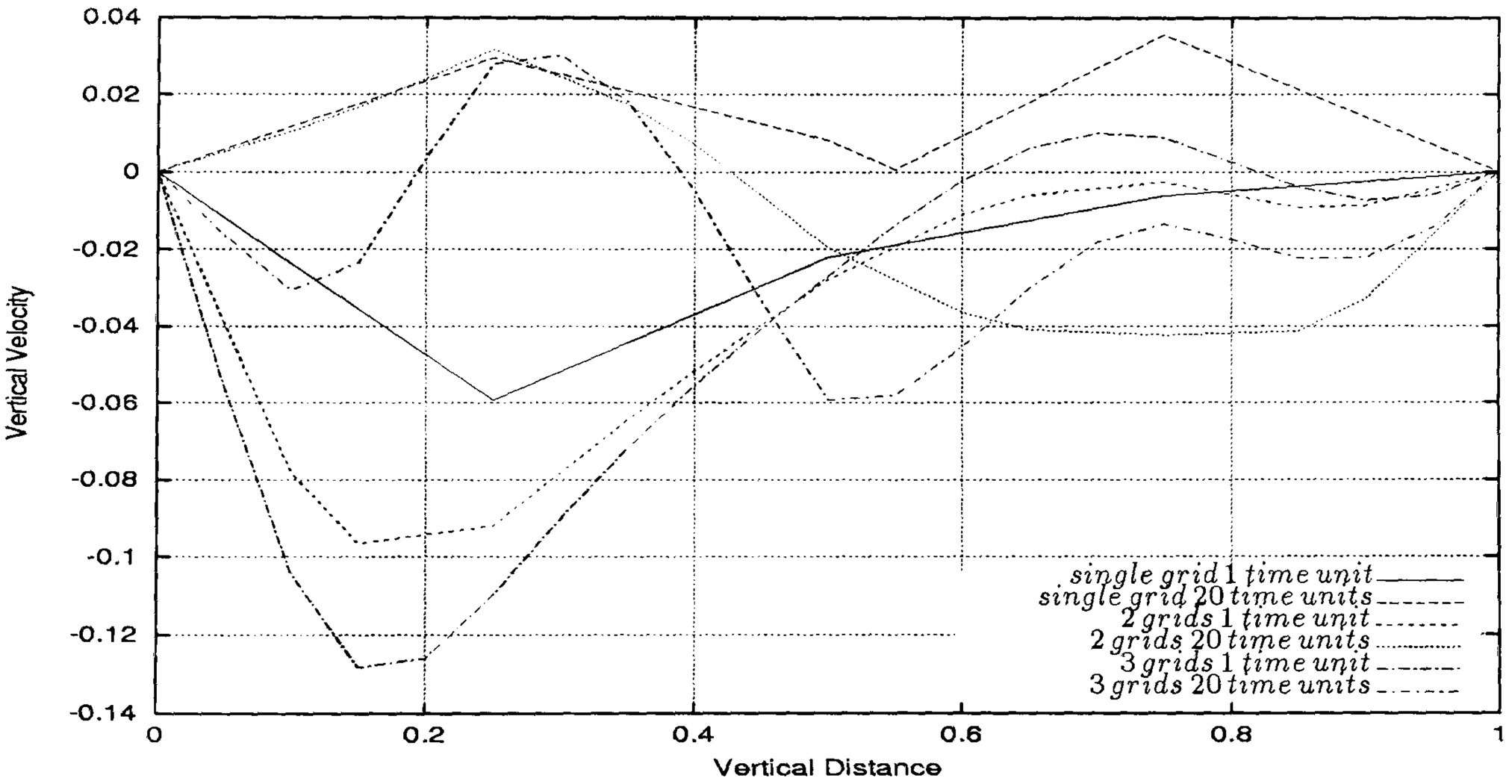


Figure 5.119: Multi-phase T-junction flow - comparison of vertical velocity profiles for *Phase II* along the line $x = 3.375$ for single 2 and 3 level computations. It can be seen that the curves for single 2 and 3 level computations at 20 time units exhibit the same pattern as at 1 time unit.

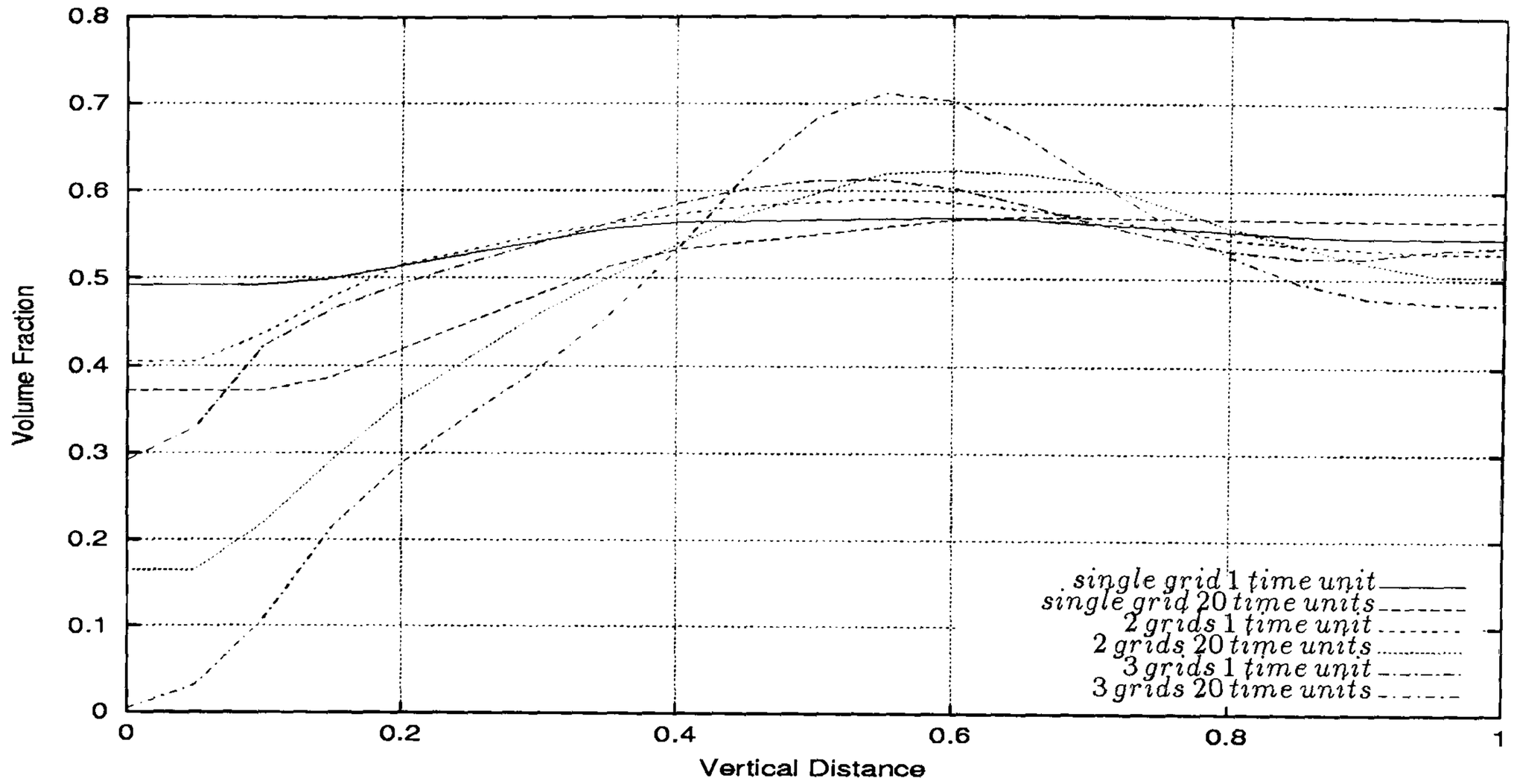


Figure 5.120: Multi-phase T-junction flow - comparison of volume fraction profiles for *Phase I* along the line $x = 3.375$ for single 2 and 3 level computations. It can be observed that for all the level calculations the volume fractions approach 0.5 at the maximum vertical distance.

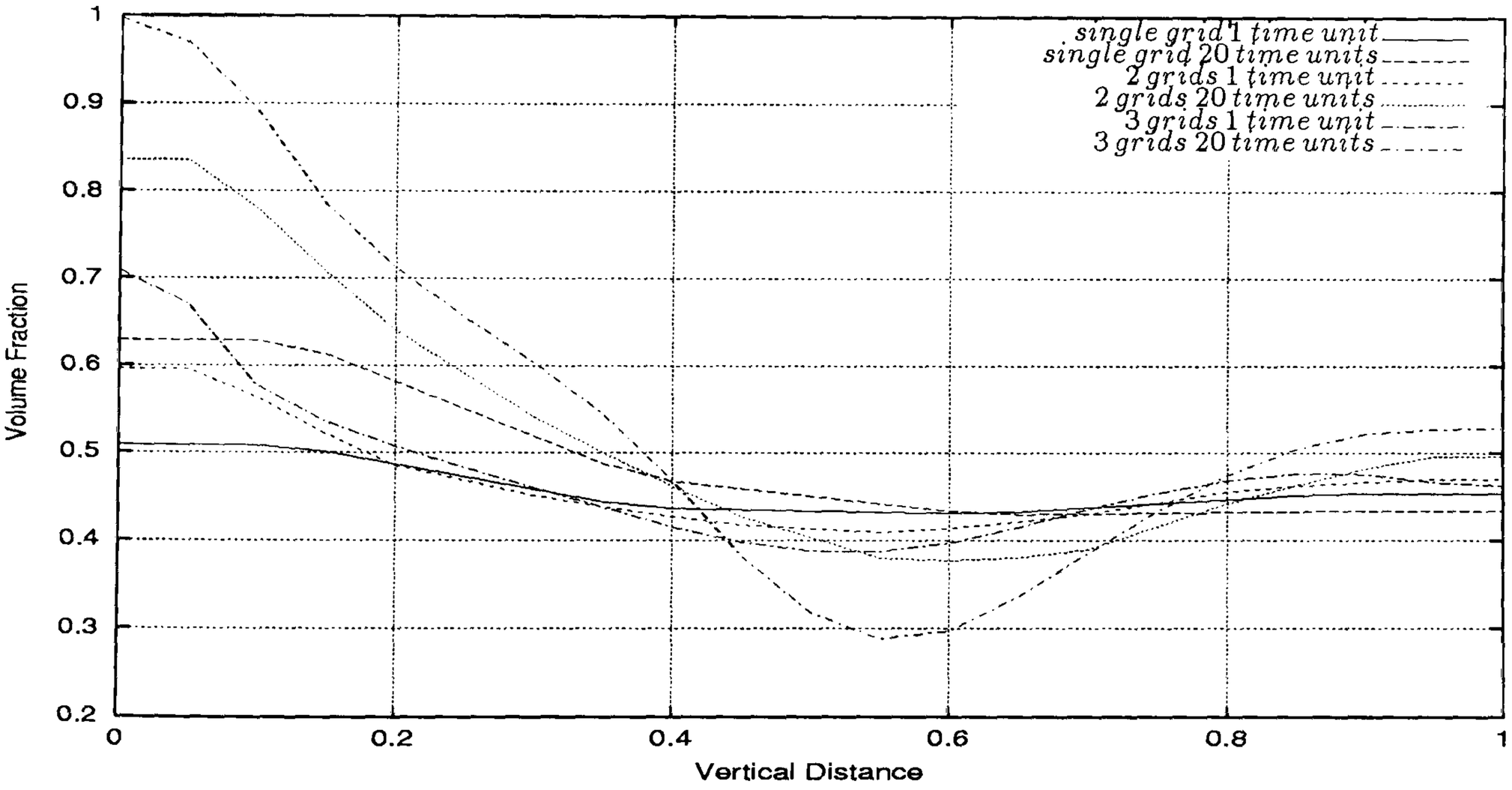


Figure 5.121: Multi-phase T-junction flow - comparison of volume fraction profiles for *Phase II* along the line $x = 3.375$ for single 2 and 3 level computations. It can be observed that for all the level calculations the volume fractions approach 0.5 at the maximum vertical distance.

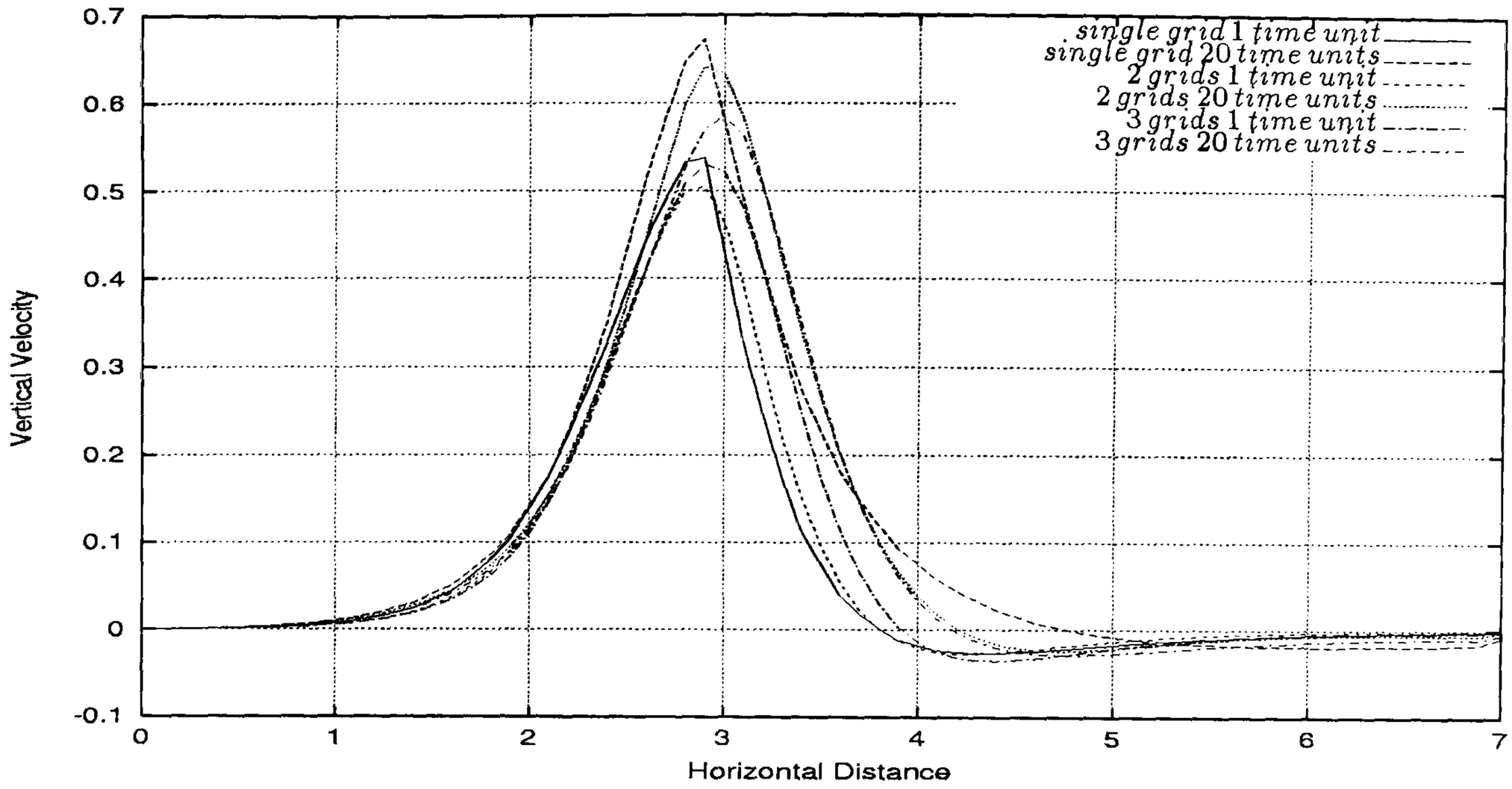


Figure 5.122: Multi-phase T-junction flow - comparison of vertical velocity profiles for *Phase I* along the line $y = 0.75$ for single 2 and 3 level computations. It can be seen that the vertical velocities reach the maximum value close to the point $x = 2.8$ for all level calculations.

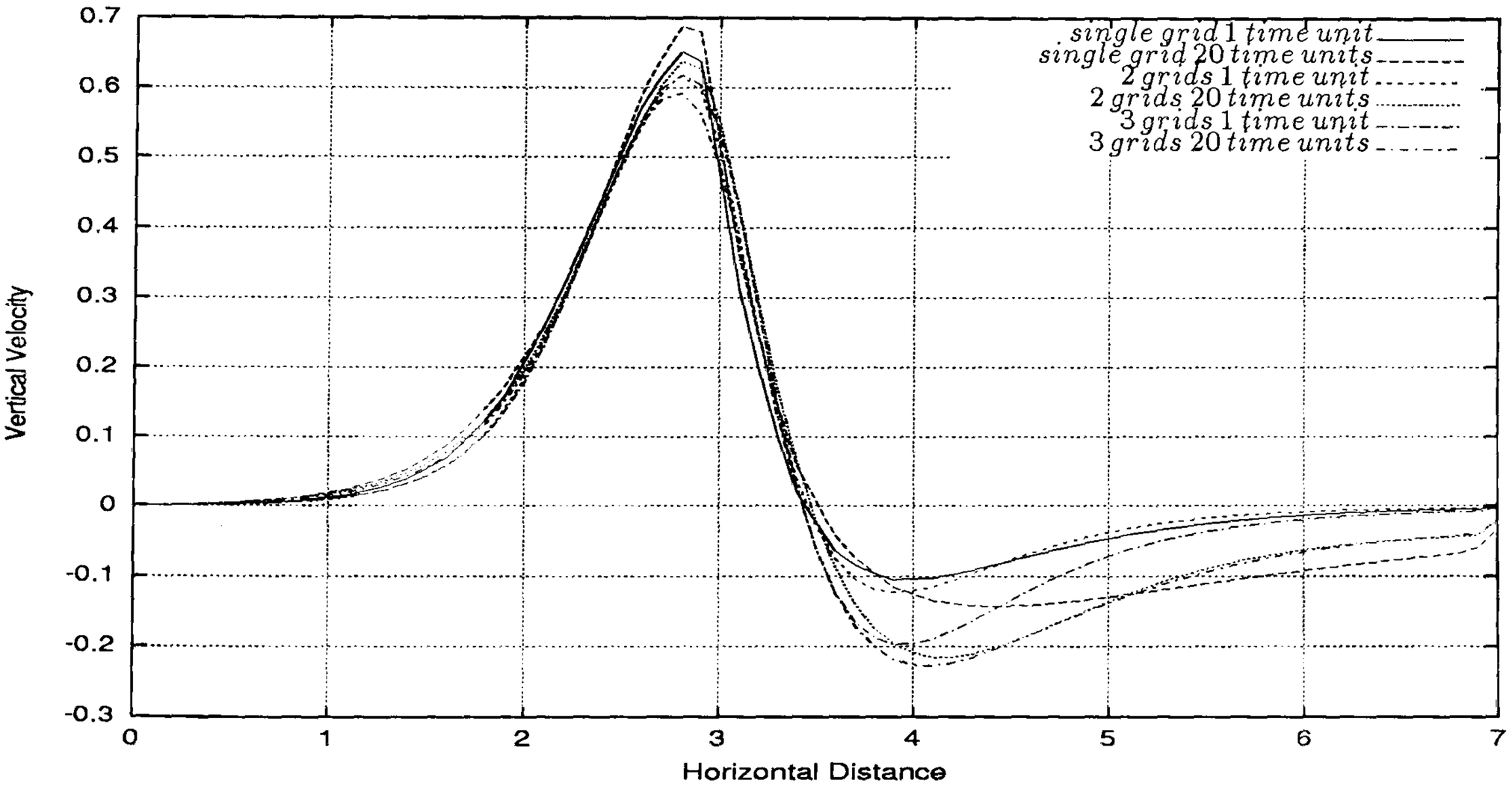


Figure 5.123: Multi-phase T-junction flow - comparison of vertical velocity profiles for *Phase II* along the line $y = 0.75$ for single 2 and 3 level computations. It can be observed that the vertical velocities reach the maximum value close to the point $x = 2.8$ for all level calculations.

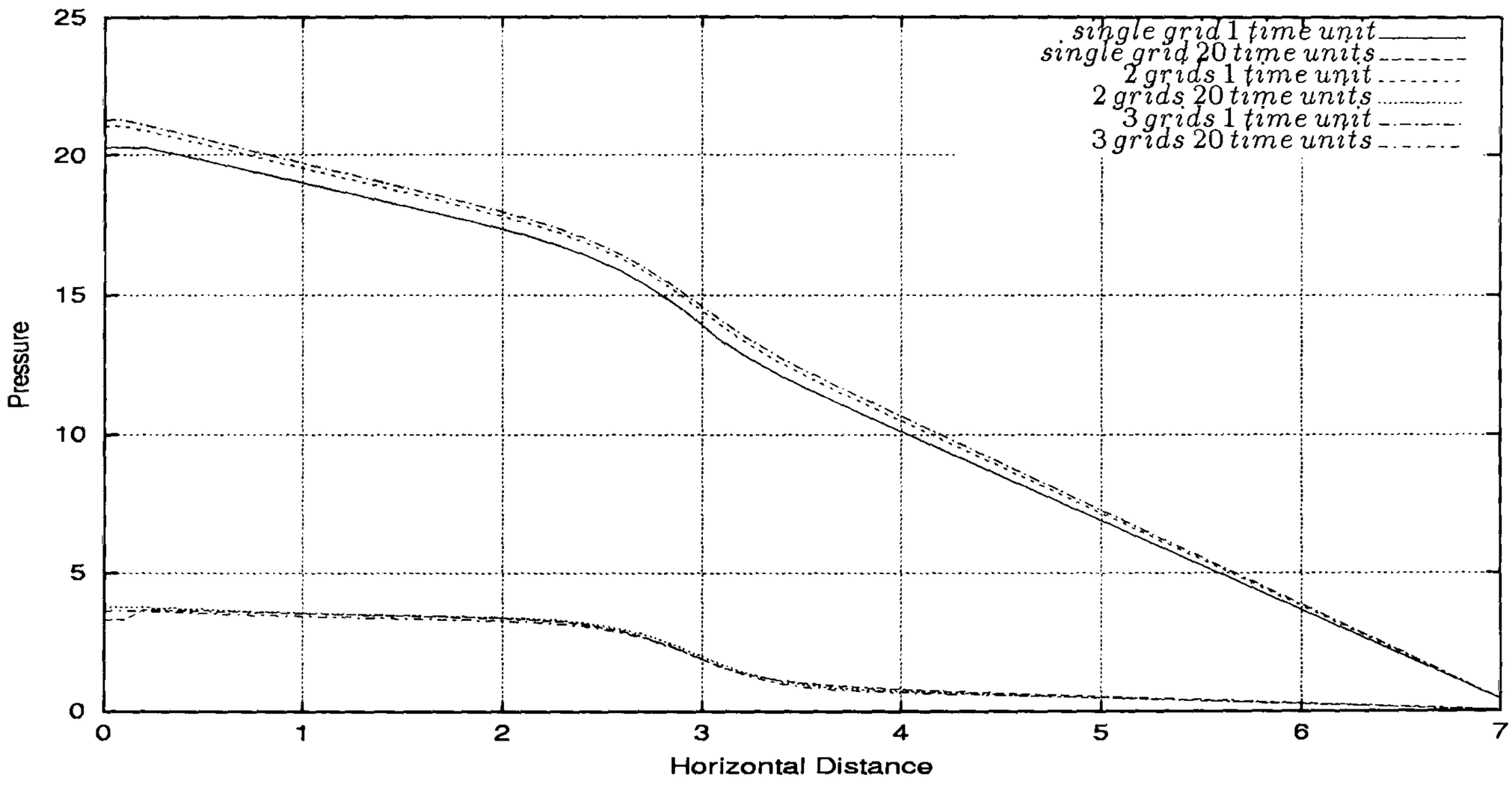


Figure 5.124: Multi-phase T-junction flow - comparison of pressure drop profiles along the line $y = 0.75$ for single 2 and 3 level computations. It can be seen good agreement between the pressure drop for all level calculations at 1 and at 20 time units.

Figures 5.125 through 5.137 present the profiles of the horizontal velocity, vertical velocity, volume fraction and pressure drop along the lines $x = 3.5$ and $y = 0.5$ for 3 level computations and two different time steps (1.0 and 10.0). In Figure 5.125 we can see the horizontal velocity profiles for *Phase I* along the line $y = 0.5$ for 3 level computations and time steps 1.0 and 10.0. It is visible that at 20 time units the curves corresponding to time steps 1.0 and 10.0 agree very well even after the horizontal distance of $x = 3.5$. Similar behaviour can be observed in Figure 5.126 which represents the horizontal velocity profiles for *Phase II* along the line $y = 0.5$.

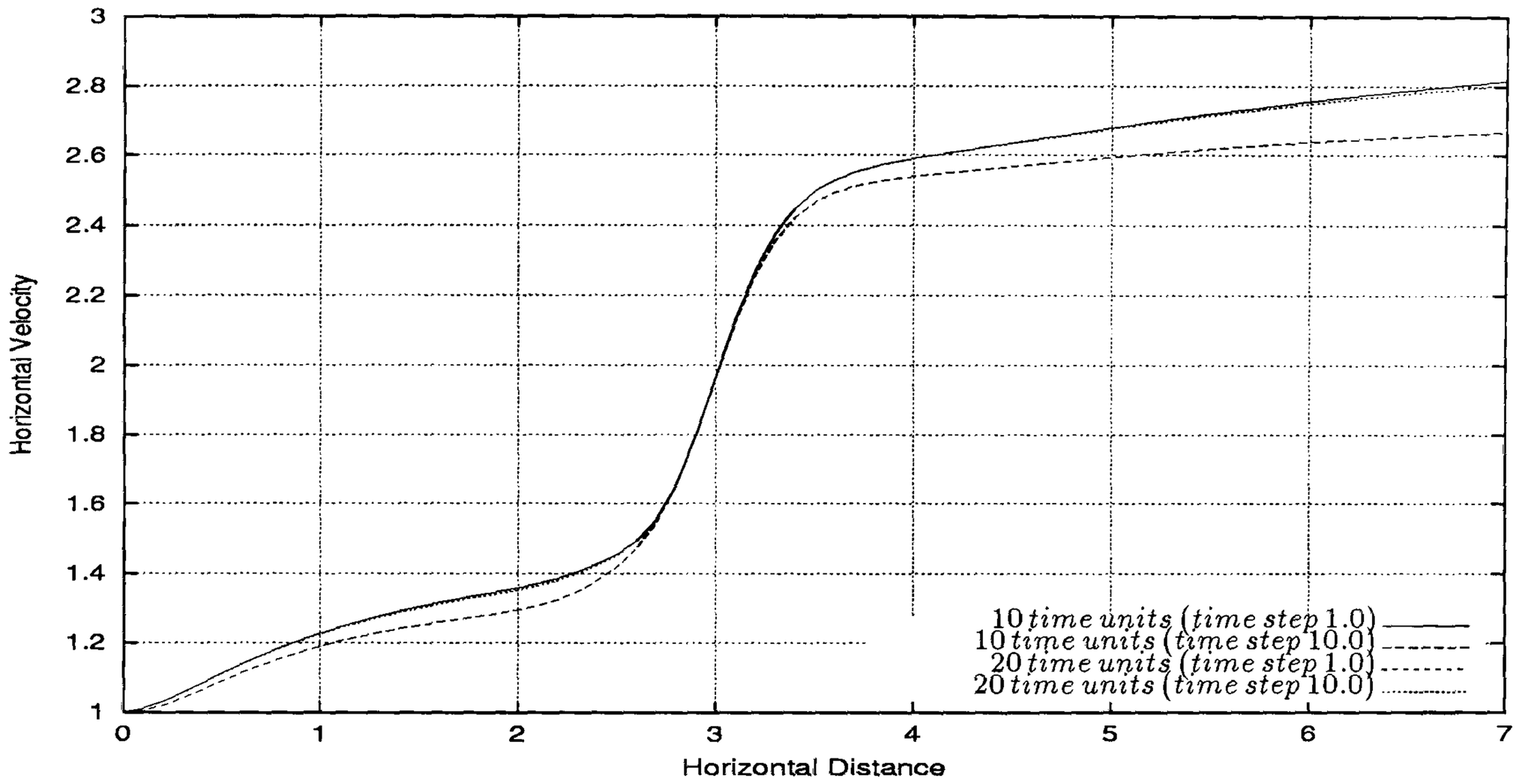


Figure 5.125: Multi-phase T-junction flow - horizontal velocity profiles for *Phase I* along the line $y = 0.5$ for 3 level computations and different time steps. It can be observed good agreement between the horizontal velocities at 20 time units for both time step calculations.

Figures 5.127 and 5.128 present the volume fraction profiles for *Phase I* and *Phase II* along the line y=0.5 for 3 level computations and with time steps 1.0 and 10.0. It can be observed the curves corresponding to the time steps 1.0 are very close to the curves corresponding to the time steps 10.0. This is particularly visible at 20 time units.

Figures 5.129 and 5.130 present the horizontal velocity profiles for *Phase I* and *Phase II* along the line $x = 3.5$ for 3 level computations with time steps 1.0 and 10.0. The agreement observed between the curves at 20 time units is good particularly for *Phase II*.

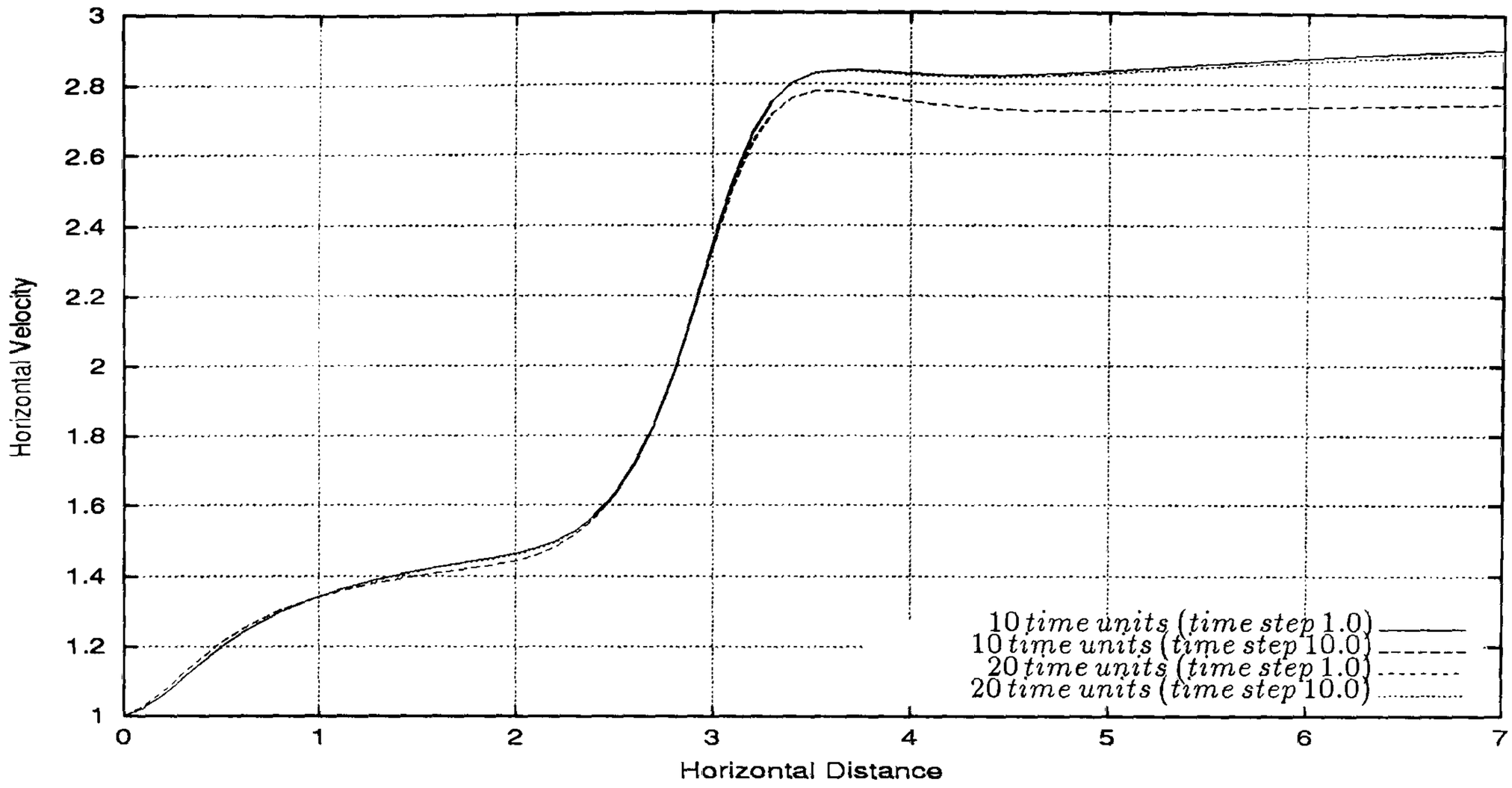


Figure 5.126: Multi-phase T-junction flow - horizontal velocity profiles for *Phase II* along the line $y = 0.5$ for 3 level computations and different time steps. It can be seen good agreement between the horizontal velocities at 20 time units for both time step calculations.

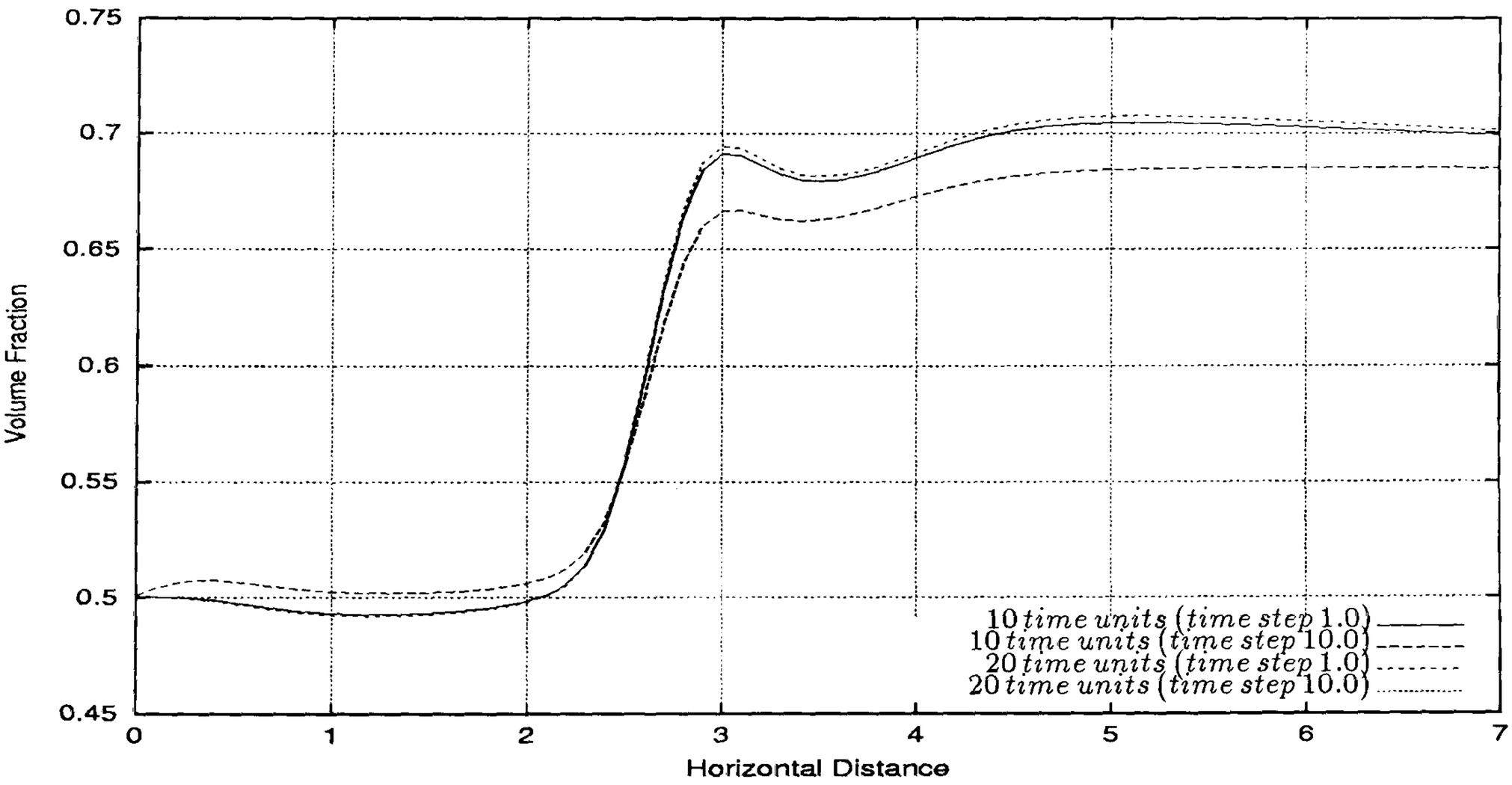


Figure 5.127: Multi-phase T-junction flow - volume fraction profiles for *Phase I* along the line $y = 0.5$ for 3 level computations and different time steps. It can be observed good agreement between the volume fractions for both time steps calculations at 20 time units.

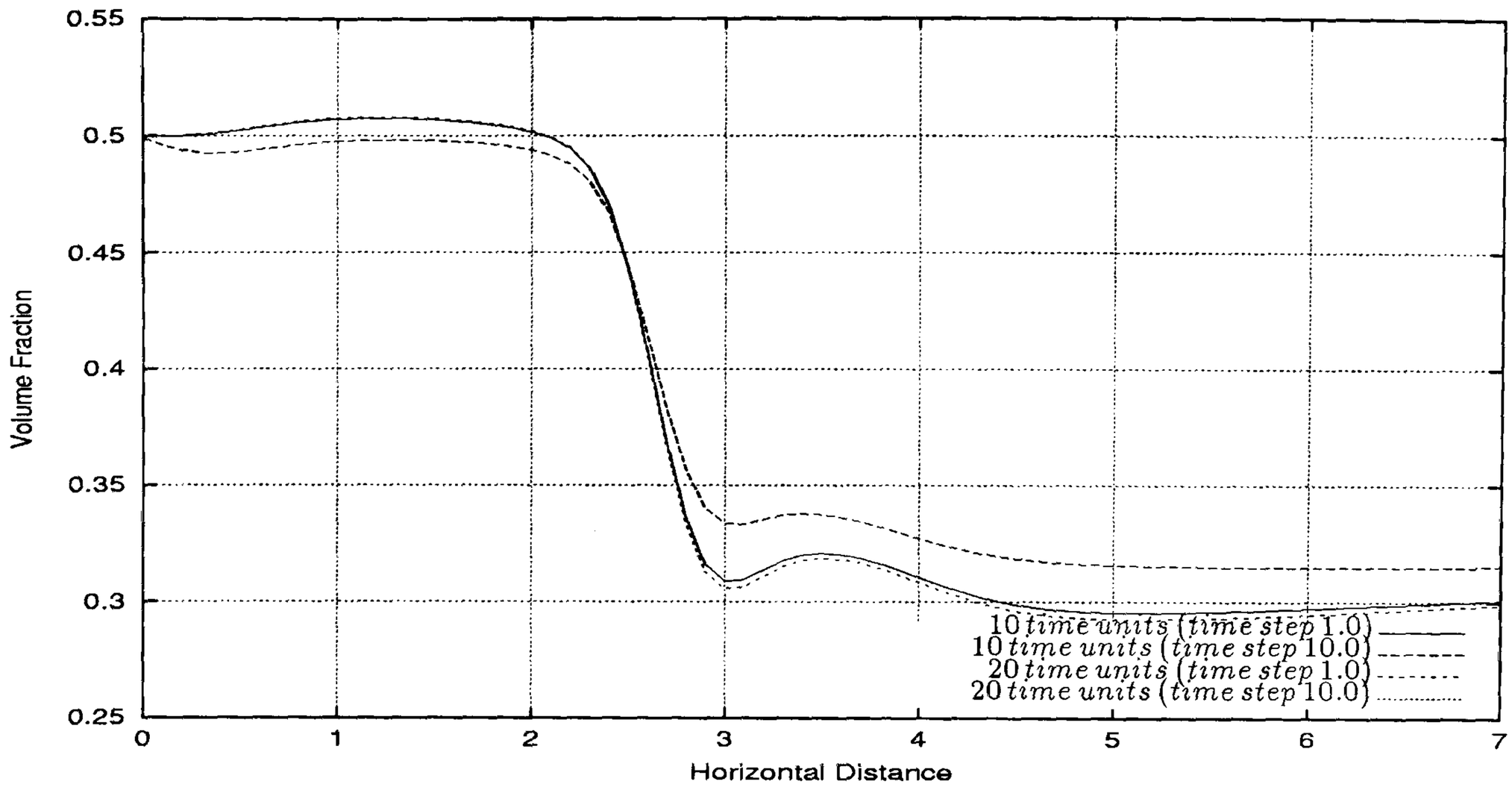


Figure 5.128: Multi-phase T-junction flow - volume fraction profiles for *Phase II* along the line $y = 0.5$ for 3 level computations and different time steps. It can be seen good agreement between the volume fractions for both time steps calculations at 20 time units.

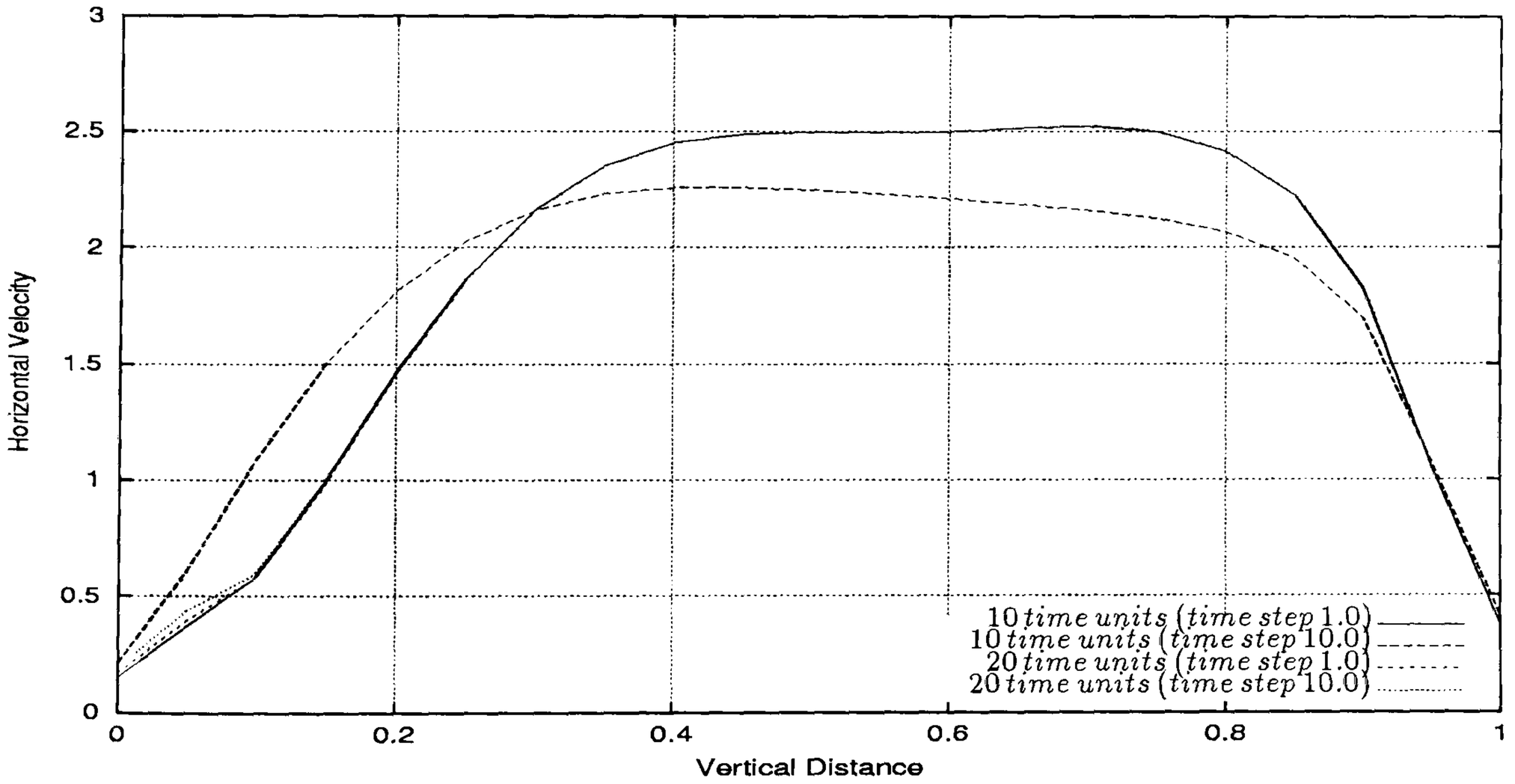


Figure 5.129: Multi-phase T-junction flow - horizontal velocity profiles for *Phase I* along the line $x = 3.5$ for 3 level computations and different time steps. It can be seen good agreement between the horizontal velocities computed by both time steps calculations at 20 time units.

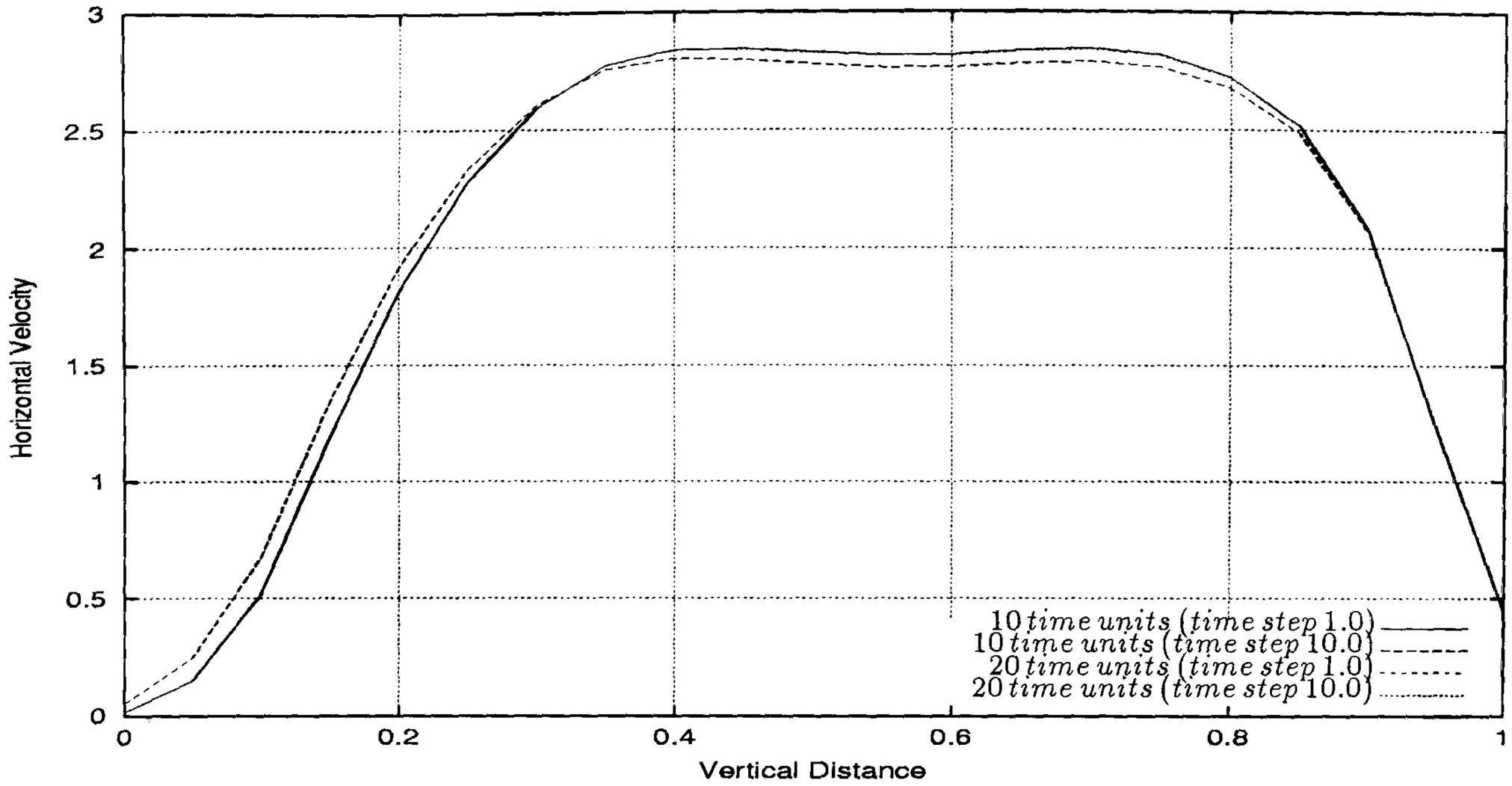


Figure 5.130: Multi-phase T-junction flow - horizontal velocity profiles for *Phase II* along the line $x = 3.5$ for 3 level computations and different time steps. It can be seen good agreement between the horizontal velocities computed by both time steps calculations.

The vertical velocity profiles for *Phase I* and *Phase II* with time step 1.0 and 10.0 are presented in Figures 5.131 and 5.132. At 20 time units the curves corresponding to time steps 1.0 and 10.0 agree very well.

In Figures 5.134 and 5.135 we present the volume fractions for both phases with time step 1.0 and 10.0 along the line $x = 3.5$. The agreement between all the entire curves is excellent at 10 and 20 time units.

In Figure 5.135 we can see the pressure drop profiles along the line $y = 0.5$ for 3 level computations and different time steps. It is visible the good agreement between all the curves.

Figures 5.136 and 5.137 present the vertical velocity profiles for *Phase I* and *Phase II* with time steps 1.0 and 10.0. The agreement between all the curves is in general good. There is a slight difference in the curves of *Phase I* between 2.8 and 4.5. However afterwards the agreement is very good.

From this study of the time step independence we can conclude that the solution algorithm exhibits a very good agreement.

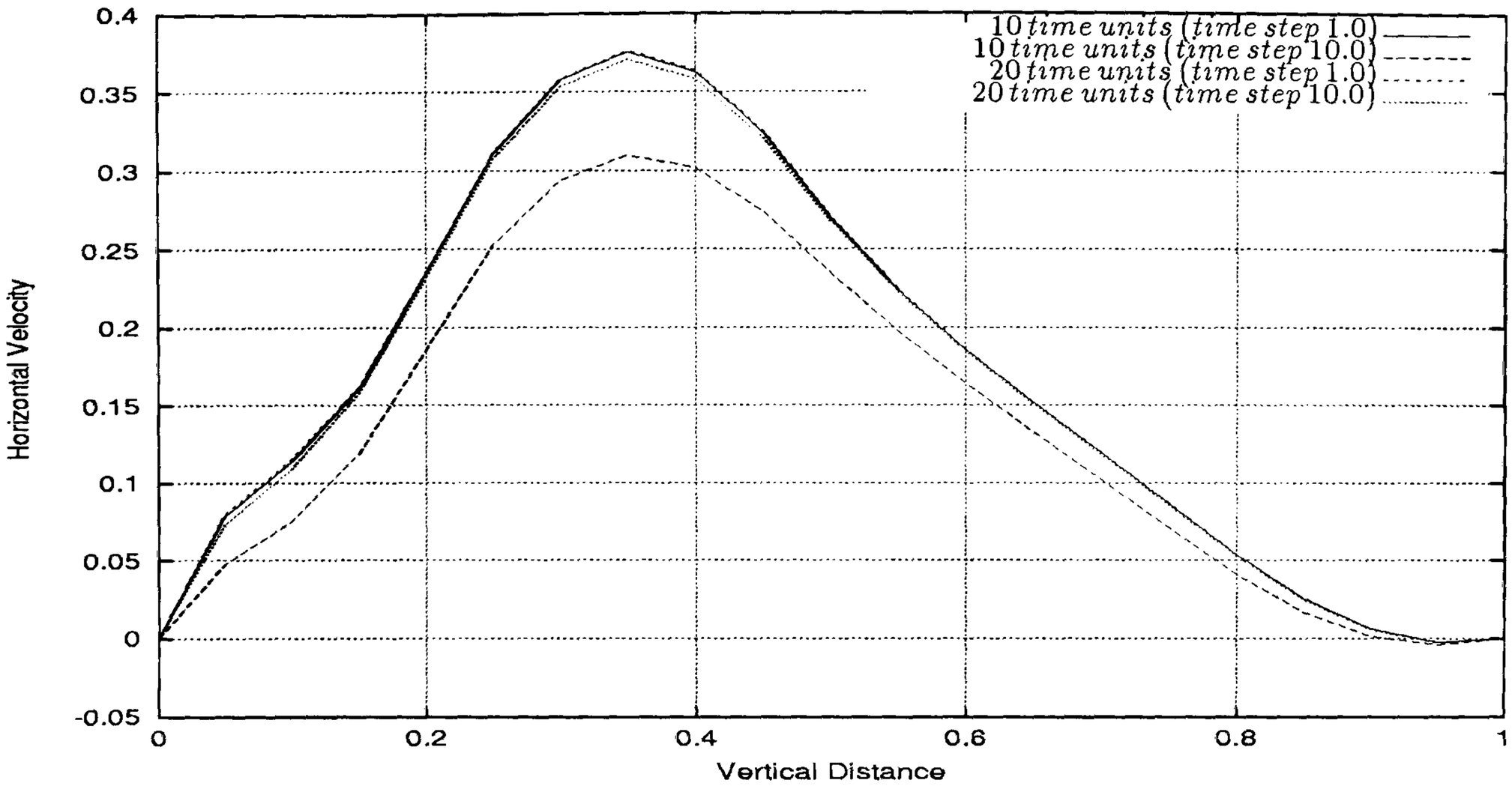


Figure 5.131: Multi-phase T-junction flow - vertical velocity profiles for *Phase I* along the line $x = 3.5$ for 3 level computations and different time steps. It can be seen good agreement between the vertical velocities computed by both time steps at 20 time units.

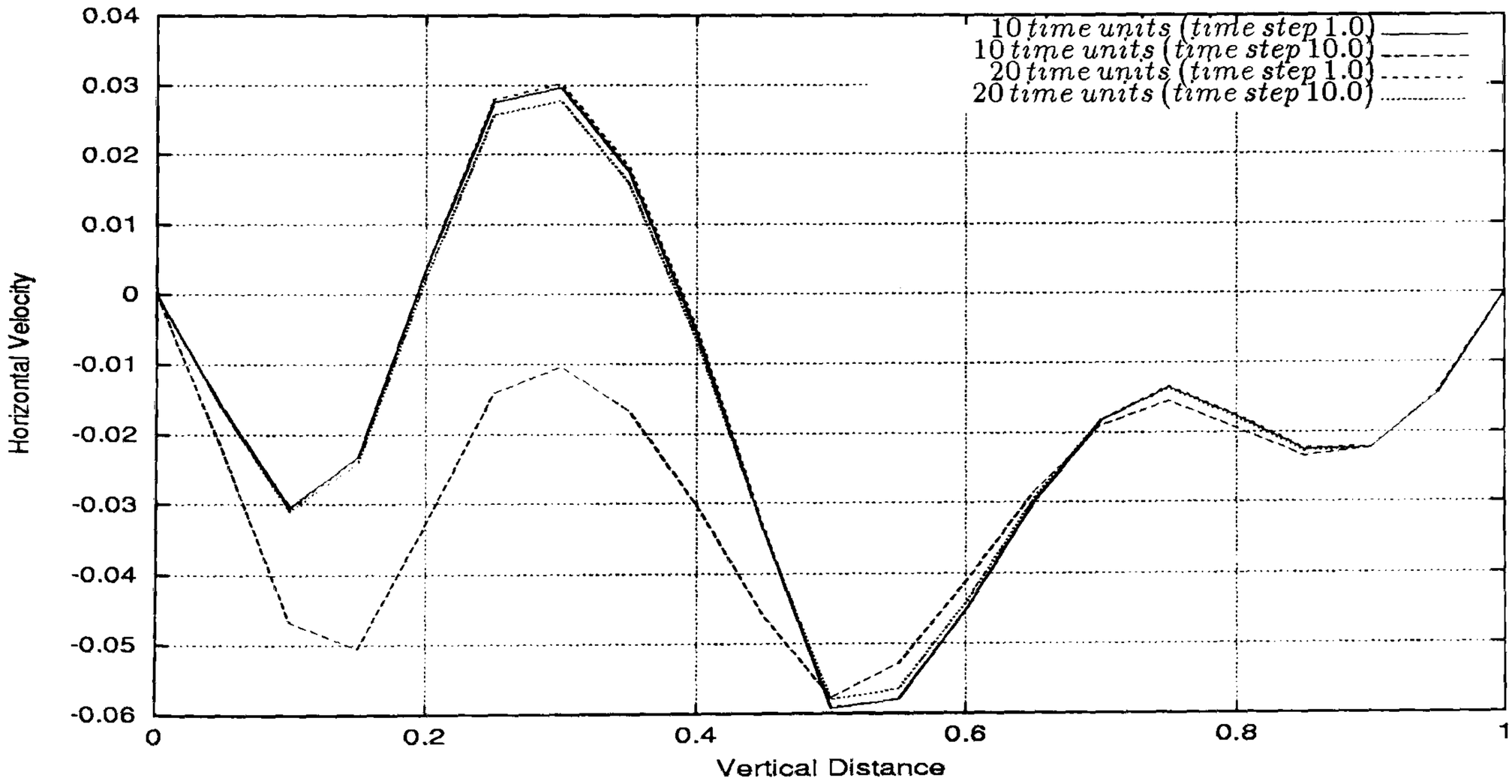


Figure 5.132: Multi-phase T-junction flow - vertical velocity profiles for *Phase II* along the line $x = 3.5$ for 3 level computations and different time steps. It can be observed good agreement between the vertical velocities computed by both time steps at 20 time units.

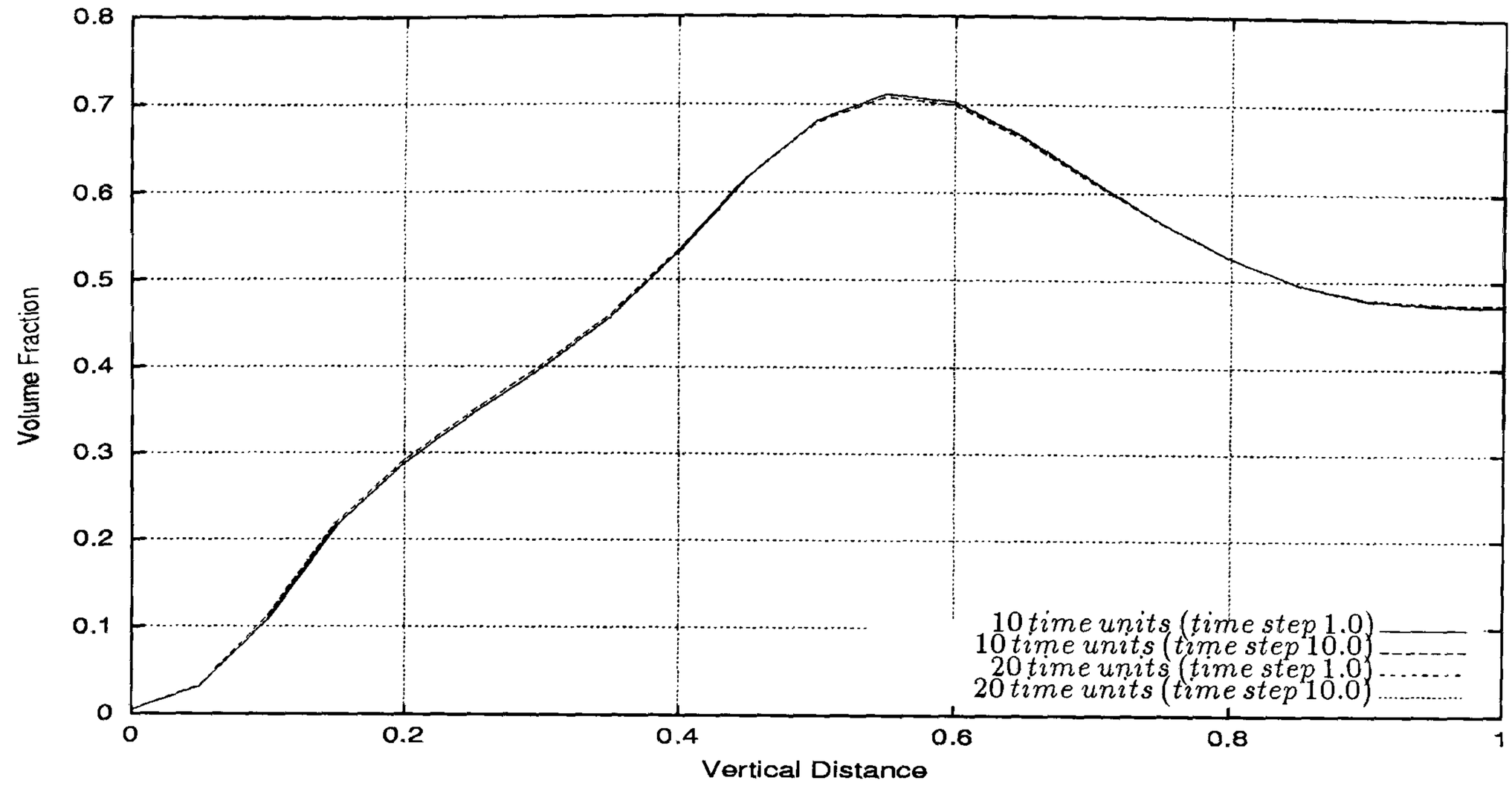


Figure 5.133: Multi-phase T-junction flow - volume fraction profiles for *Phase I* along the line $x = 3.5$ for 3 level computations and different time steps. It can be seen excellent agreement between the volume fractions computed by both time steps.

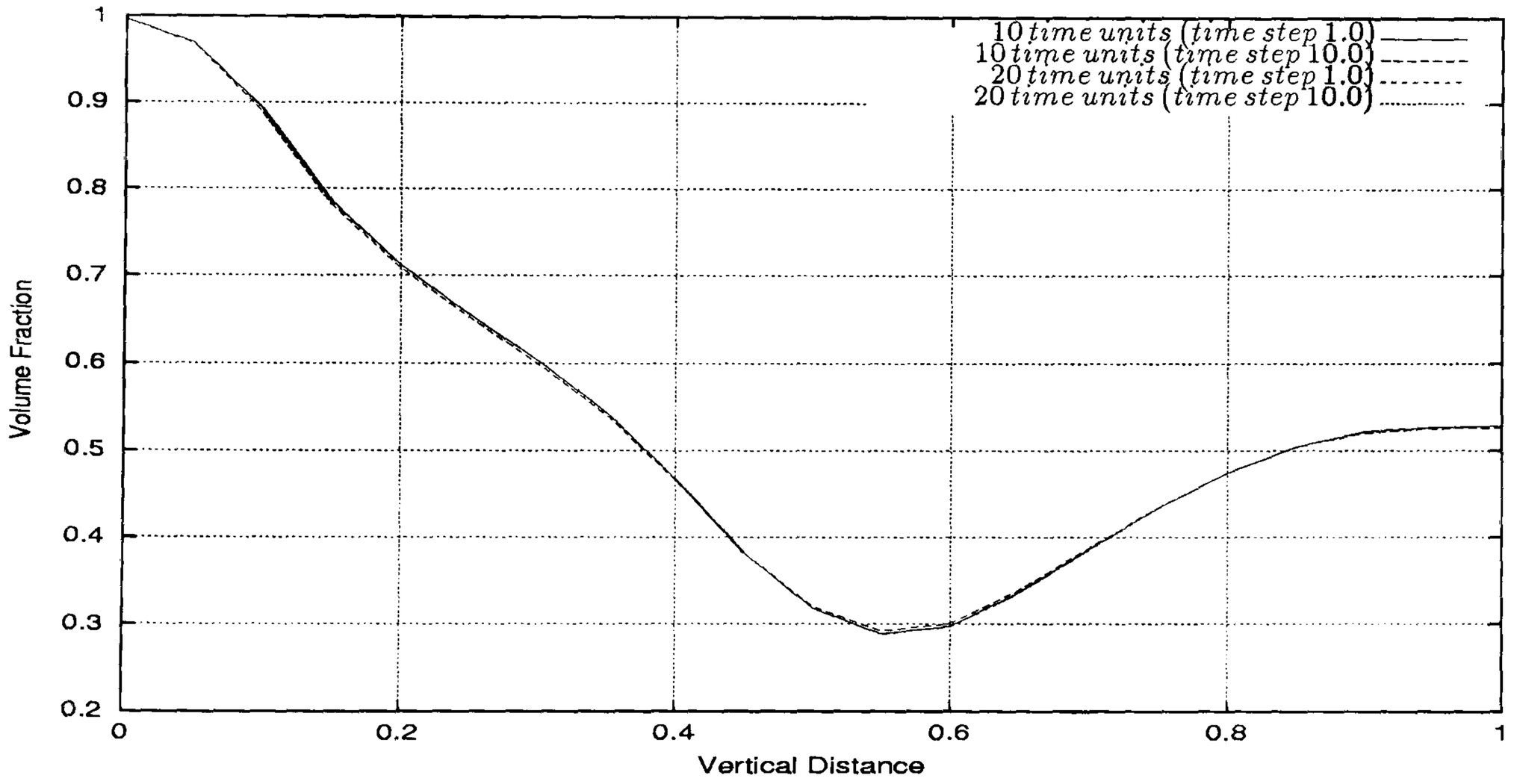


Figure 5.134: Multi-phase T-junction flow - volume fraction profiles for *Phase II* along the line $x = 3.5$ for 3 level computations and different time steps. It can be seen excellent agreement between the volume fractions computed by both time steps.

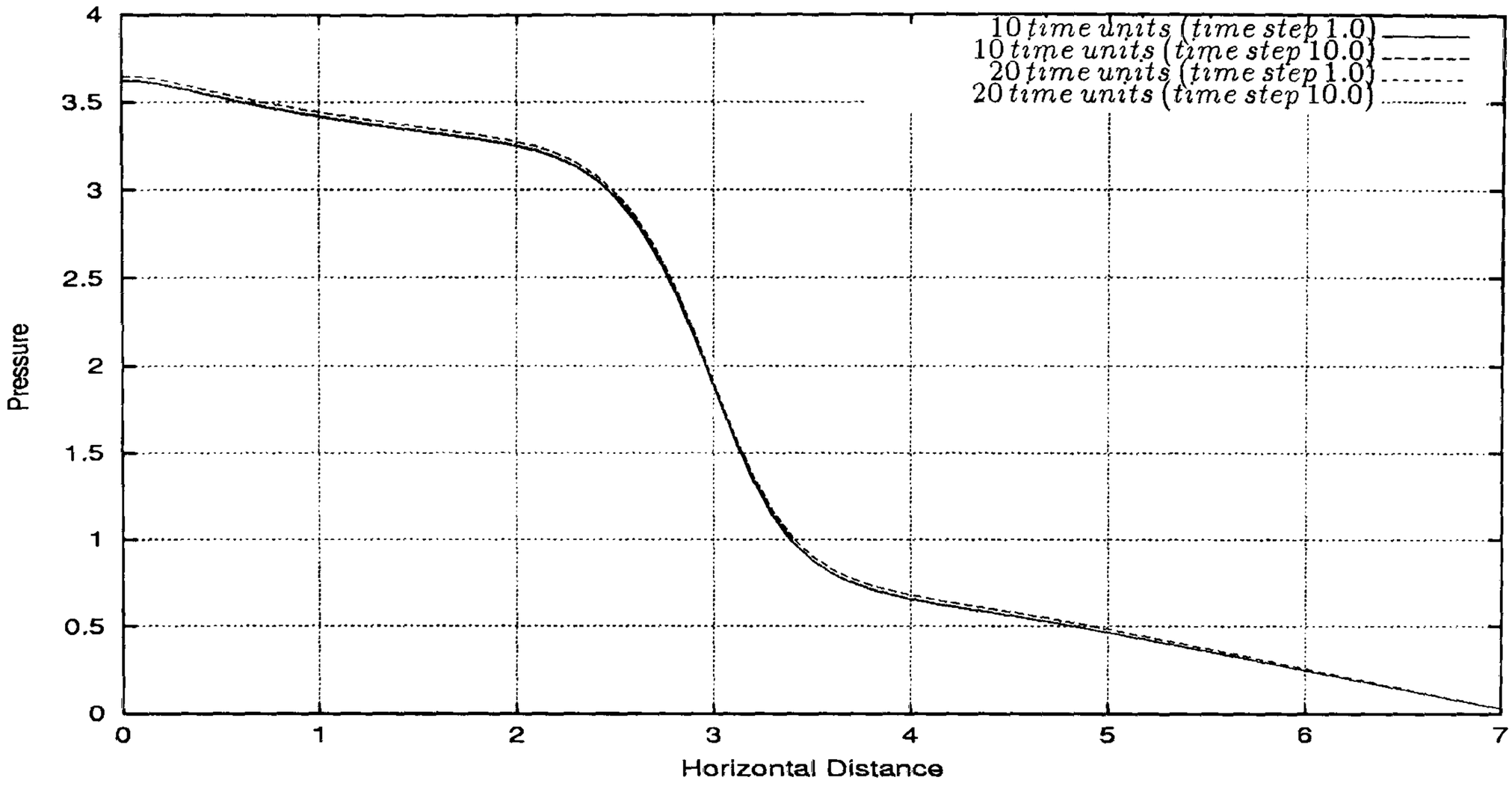


Figure 5.135: Multi-phase T-junction flow - pressure drop profiles along the line $y = 0.5$ for 3 level computations and different time steps. It can be observed good agreement between the pressure drop values produced by both time step calculations.

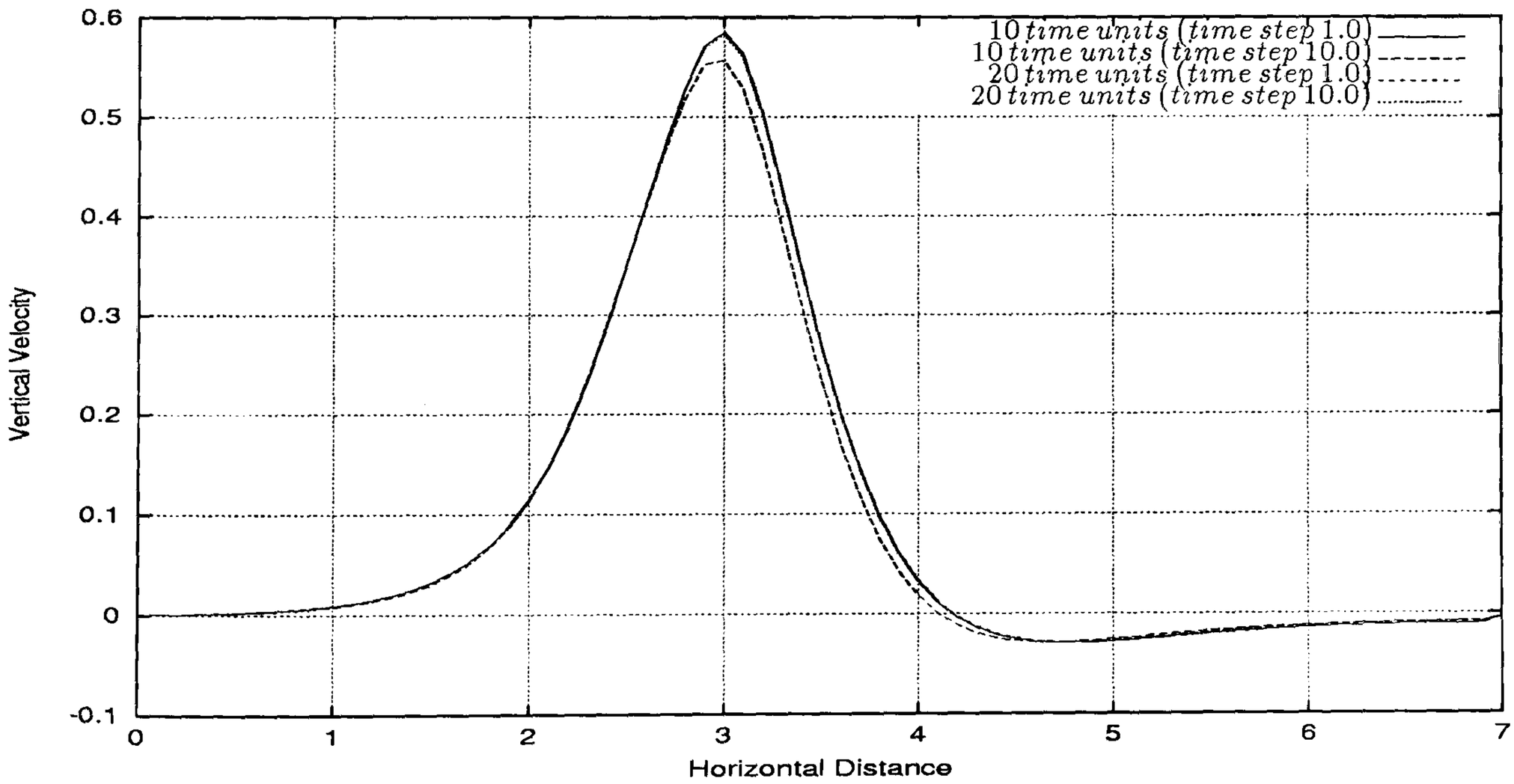


Figure 5.136: Multi-phase T-junction flow - vertical velocity profiles for *Phase I* along the line $y = 0.5$ for 3 level computations and different time steps. It can be seen good agreement between the vertical velocities computed by both time steps calculations.

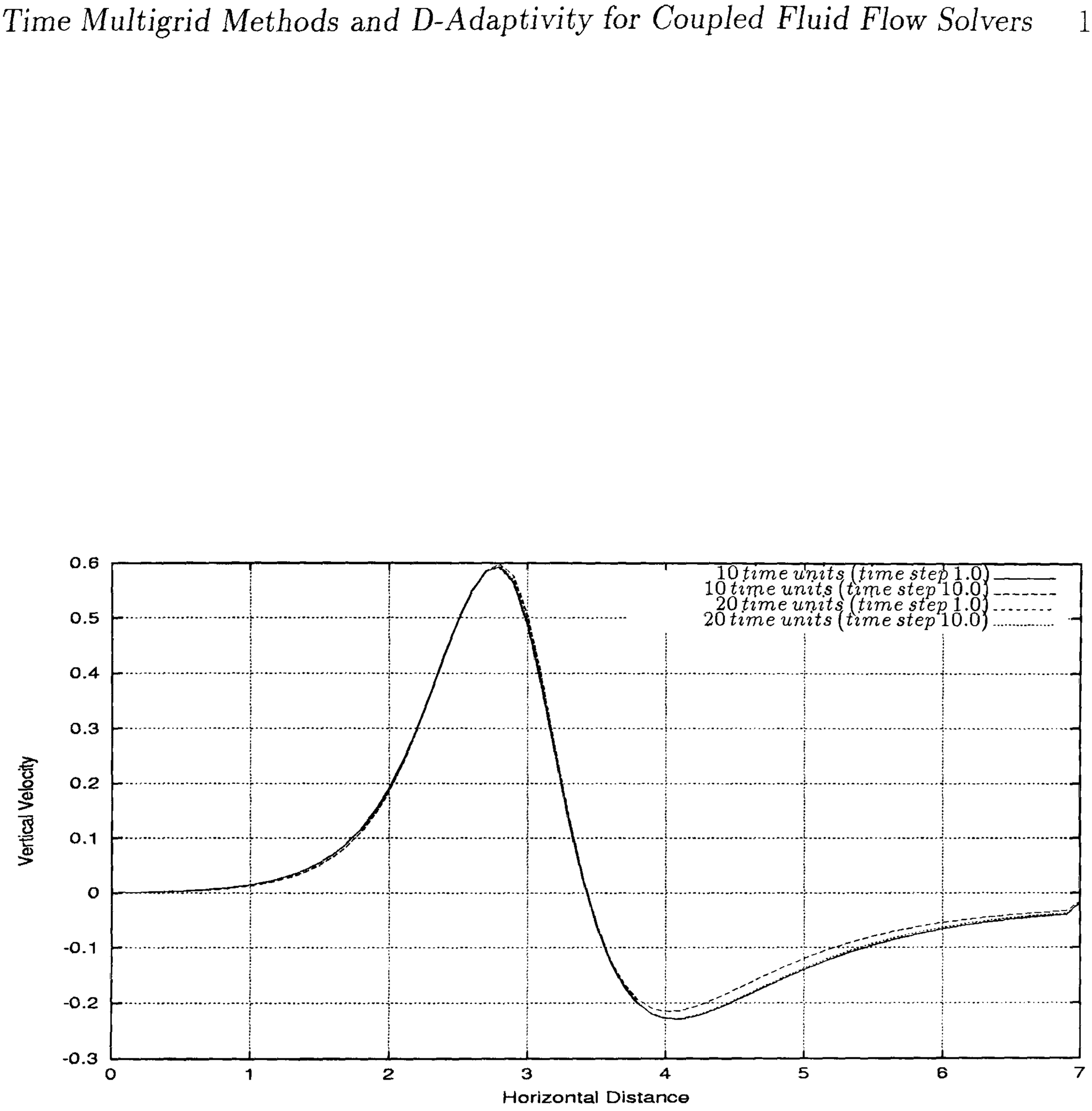


Figure 5.137: Multi-phase T-junction flow - vertical velocity profiles for *Phase II* along the line $y = 0.5$ for 3 level computations and different time steps. It can be observed good agreement between the vertical velocities computed by both time steps calculations.

Comparison with CFX 4.3 Solutions

Figures 5.138 through 5.144 present the profiles of the horizontal velocity, vertical velocity, volume fraction and pressure drop along the lines $x = 3.5$ and $y = 0.5$ for `transient Pamg Multi-phase` and `CFX 4.3`. In Figure 5.138 we present the horizontal velocity along the line $y = 0.5$ produced by `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It is visible that for both phases the curves corresponding to `CFX 4.3` and `transient Pamg Multi-phase` are very close each other.

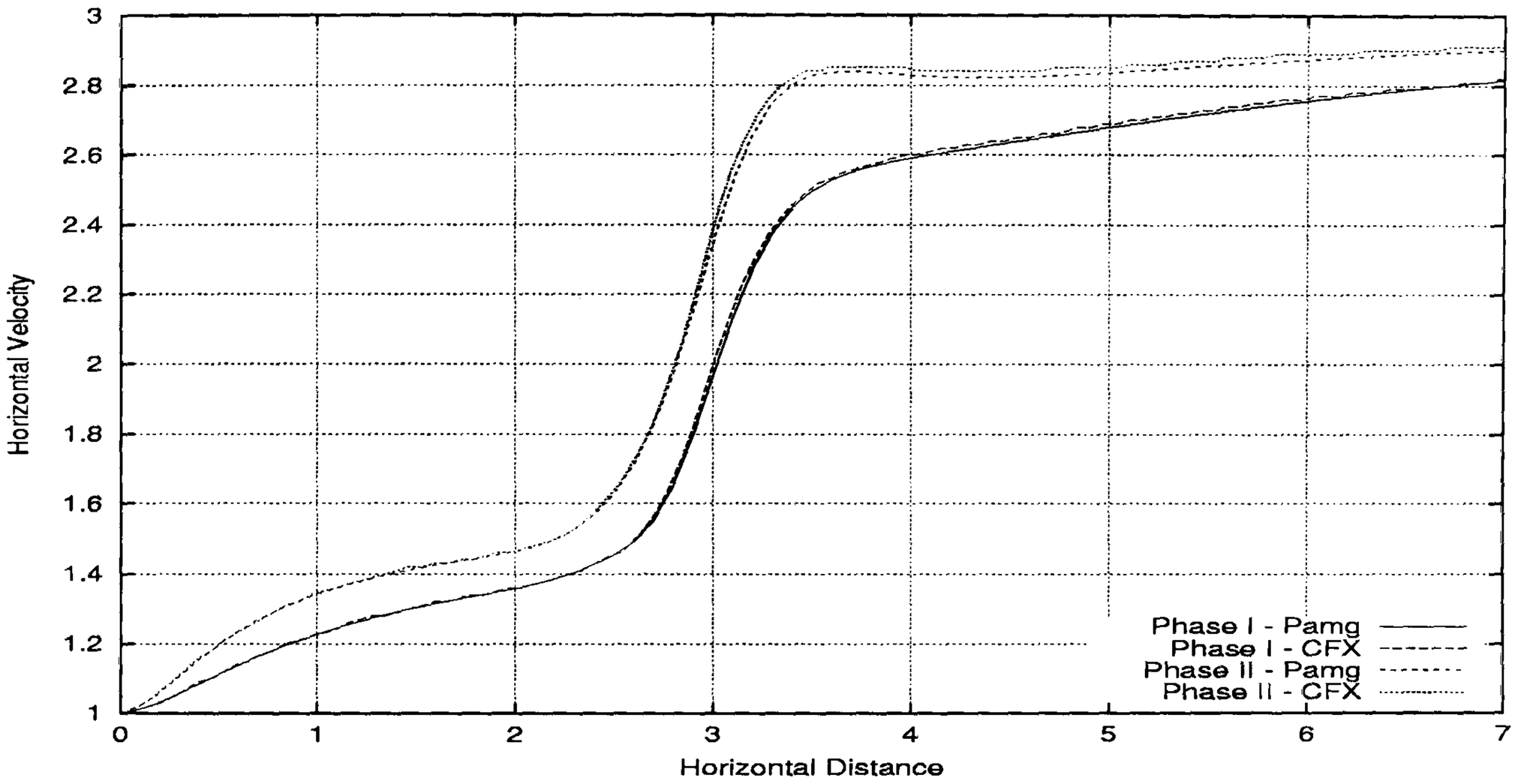


Figure 5.138: Multi-phase T-junction flow - comparison of horizontal velocity along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen good agreement between the horizontal velocities computed by both solvers.

Figure 5.139 present the volume fraction along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. Good agreement is observed until the horizontal distance of 2.7. Afterwards there are slight differences between the values of volume fractions produced by `transient Pamg Multi-phase` and `CFX 4.3`. However the pattern that can be seen is very similar.

In Figure 5.140 we show the horizontal velocity along the line $x = 3.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. Like the curves representing the horizontal velocity produced by `transient Pamg Multi-phase`, the curves produced by `CFX 4.3` are not entirely symmetric. However the agreement between the curves produced by both solvers is good.

Figure 5.141 shows the vertical velocity along the line $x = 3.5$ produced by `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. The differences observed particularly in the vertical velocity until the vertical distance of 0.3 must be due to the different location of the velocities in the cells. While in `CFX 4.3` all the unknowns are computed in the centre of the cells, in `transient Pamg Multi-phase`

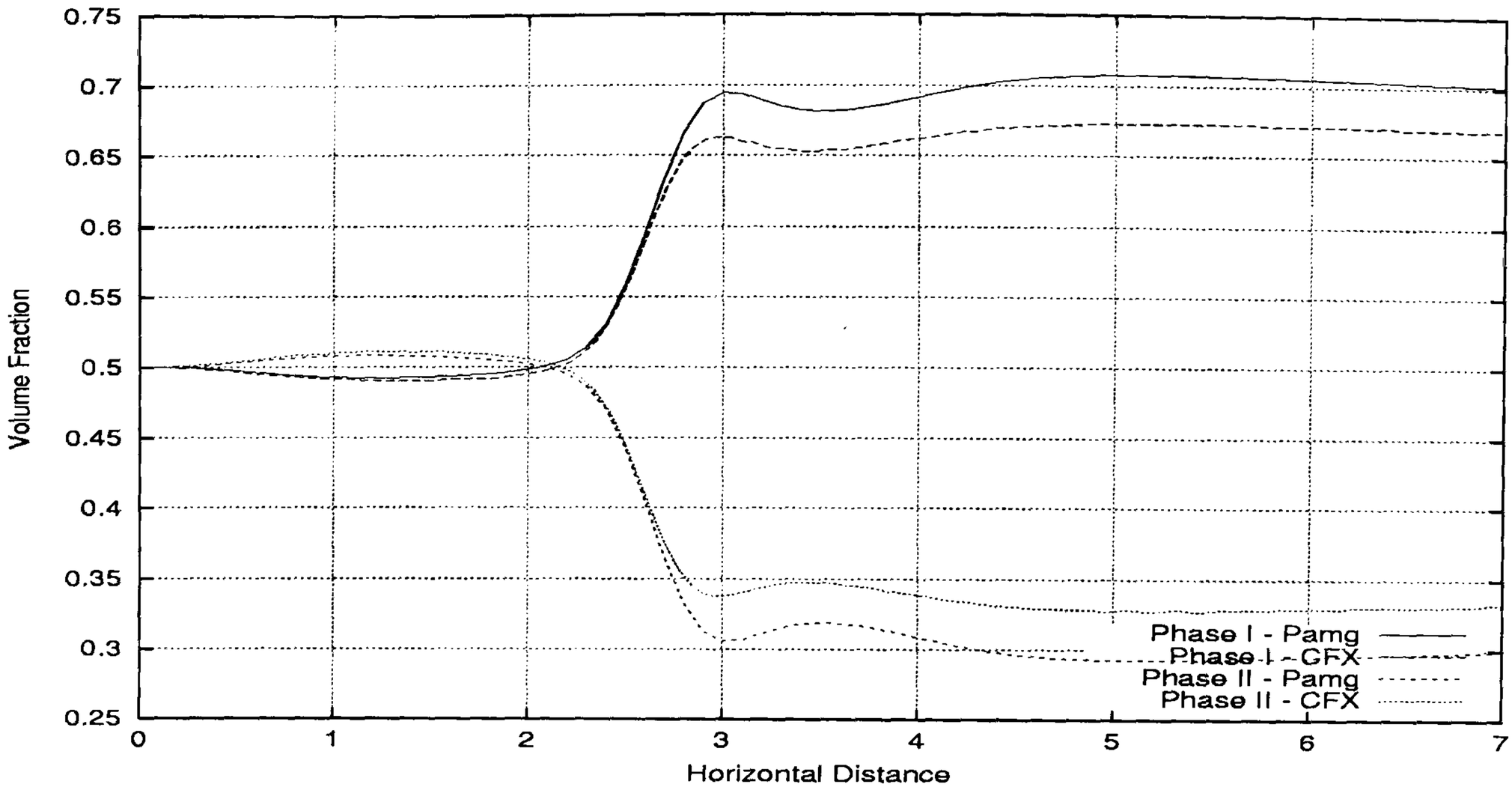


Figure 5.139: Multi-phase T-junction flow - comparison of volume fraction along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be observed the same pattern on the profiles of the volume fractions produced by both solvers.

the velocities are computed in the edges. Consequently it is expected that in a difficult region like the line $x = 3.5$ that differences will become more evident.

In Figure 5.142 we present the volume fractions along the line $x = 3.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. The difference observed in the inlet is naturally due to the initial conditions used by `CFX 4.3`, which are settled to be 0.5 for both phases.

Figure 5.143 presents the vertical velocity along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. Good agreement is observed between the curves produced by both solvers.

Finally in Figure 5.144 we present the pressure drop along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It is visible that the agreement obtained by both solvers is excellent.

Summary

To continue the validation process of the `transient Pamg Multi-phase` we chose another problem substantially more difficult than the two-phase channel flow. In this problem a two-phase fluid flow through a T-junction with two inlets is considered. Two-different fluids with different viscosities and densities have been used in this simulation. Particularly important sections have been chosen to present the solution profiles. 3 level computations have been showed. The grid and time step independence of the solutions has been verified. This has been established by comparing results for single 2 and 3 level computations and for time steps 1.0 and 10.0.

To complement the validation process results produced by `CFX 4.3` have been

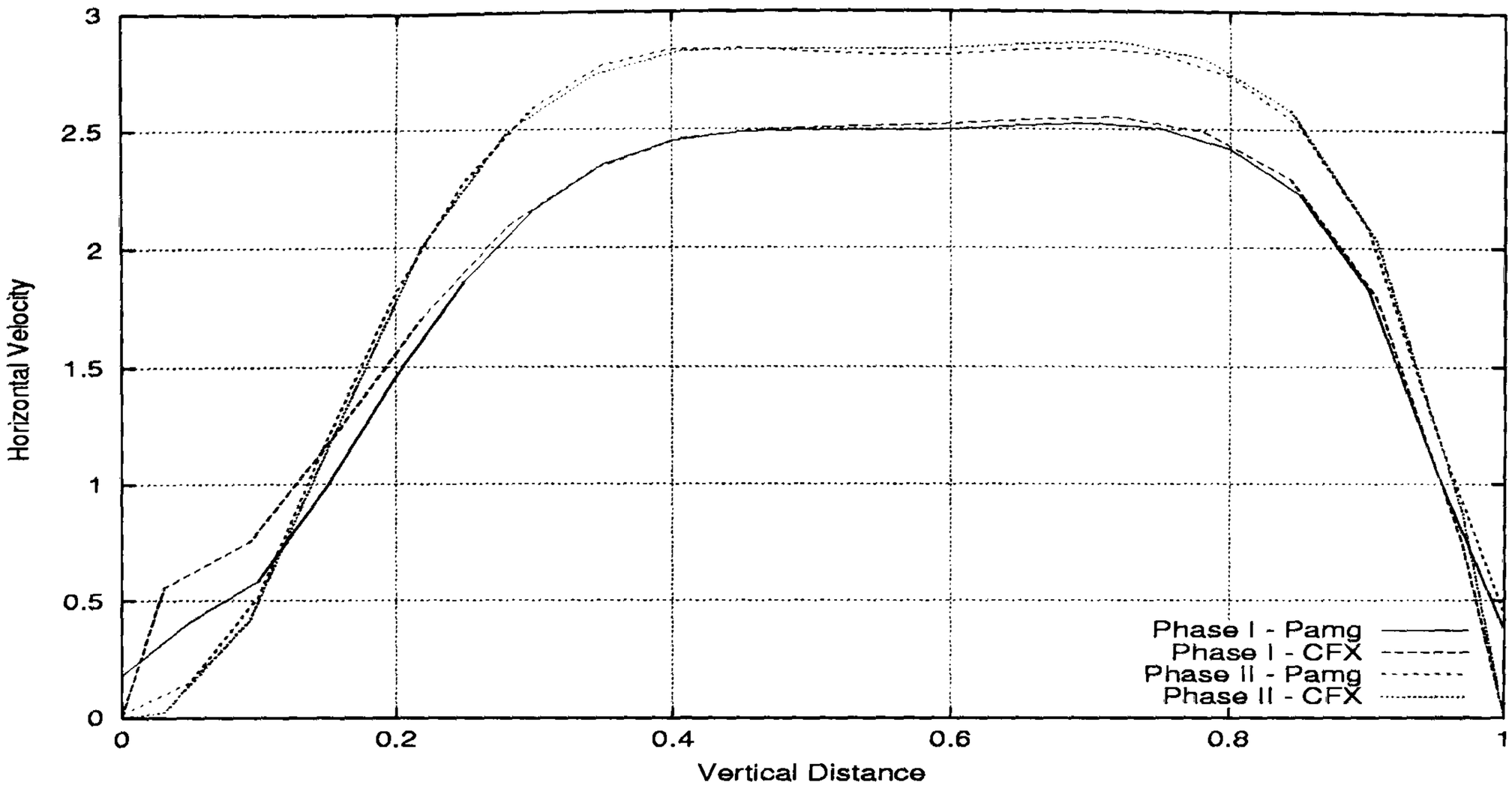


Figure 5.140: Multi-phase T-junction flow - comparison of horizontal velocity along the line $x = 3.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen good general agreement between the parabolic profiles produced by both solvers.

compared with the results produced by the `transient Pamg Multi-phase` solver. Generally we observed very good agreement between the results produced by both solvers. Slight differences observed are due to two main reasons. First of all the different location of the unknowns in the control volume. In `CFX 4.3` the cells are always cell-centred. Secondly the initialisation conditions for the volume fractions in the `transient Pamg Multi-phase` and `CFX 4.3` are slightly different.

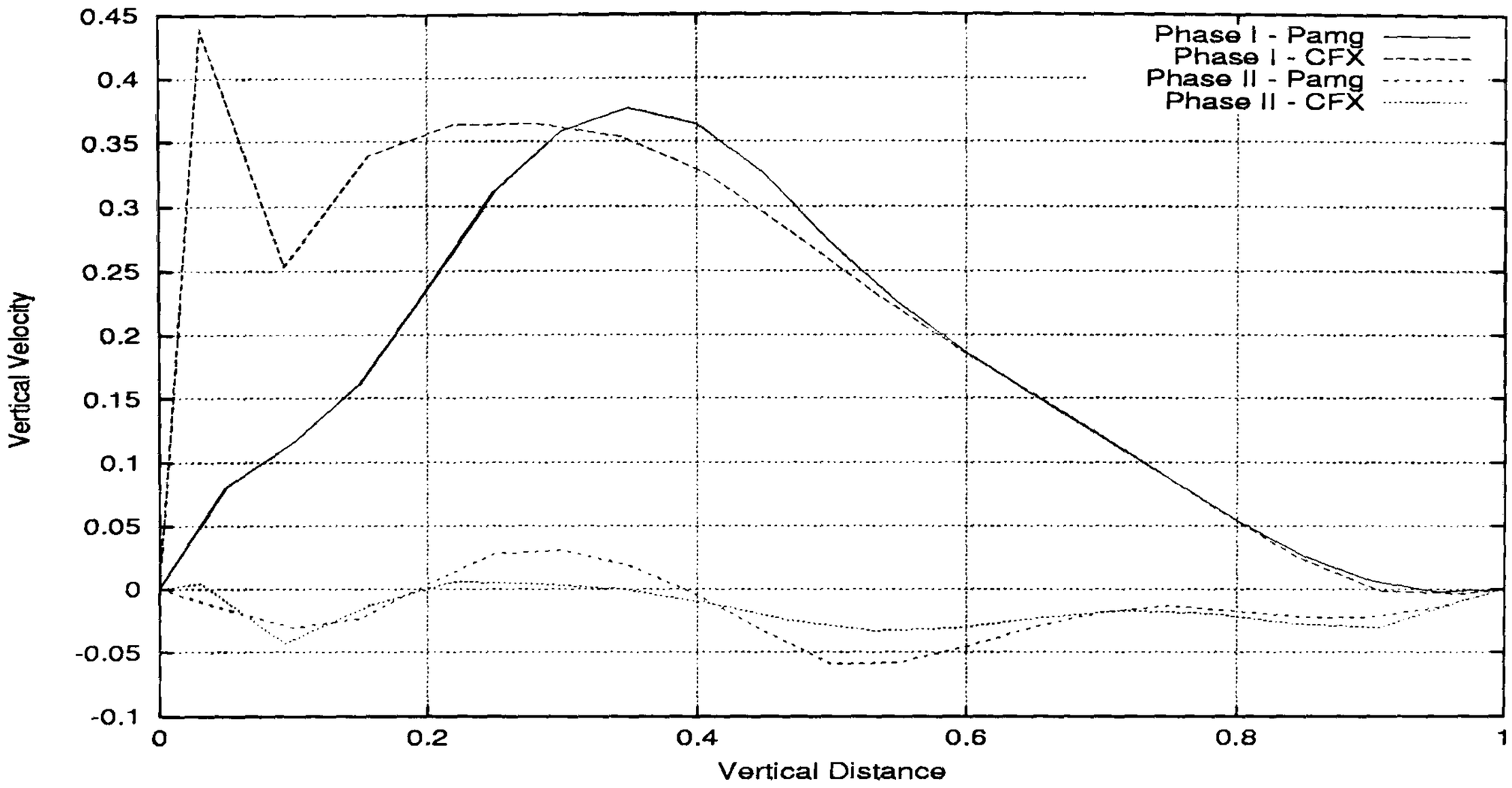


Figure 5.141: Multi-phase T-junction flow - comparison of vertical velocity along the line $x = 3.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen important differences between the vertical velocities for *Phase I* near the wall $y = 0.0$.

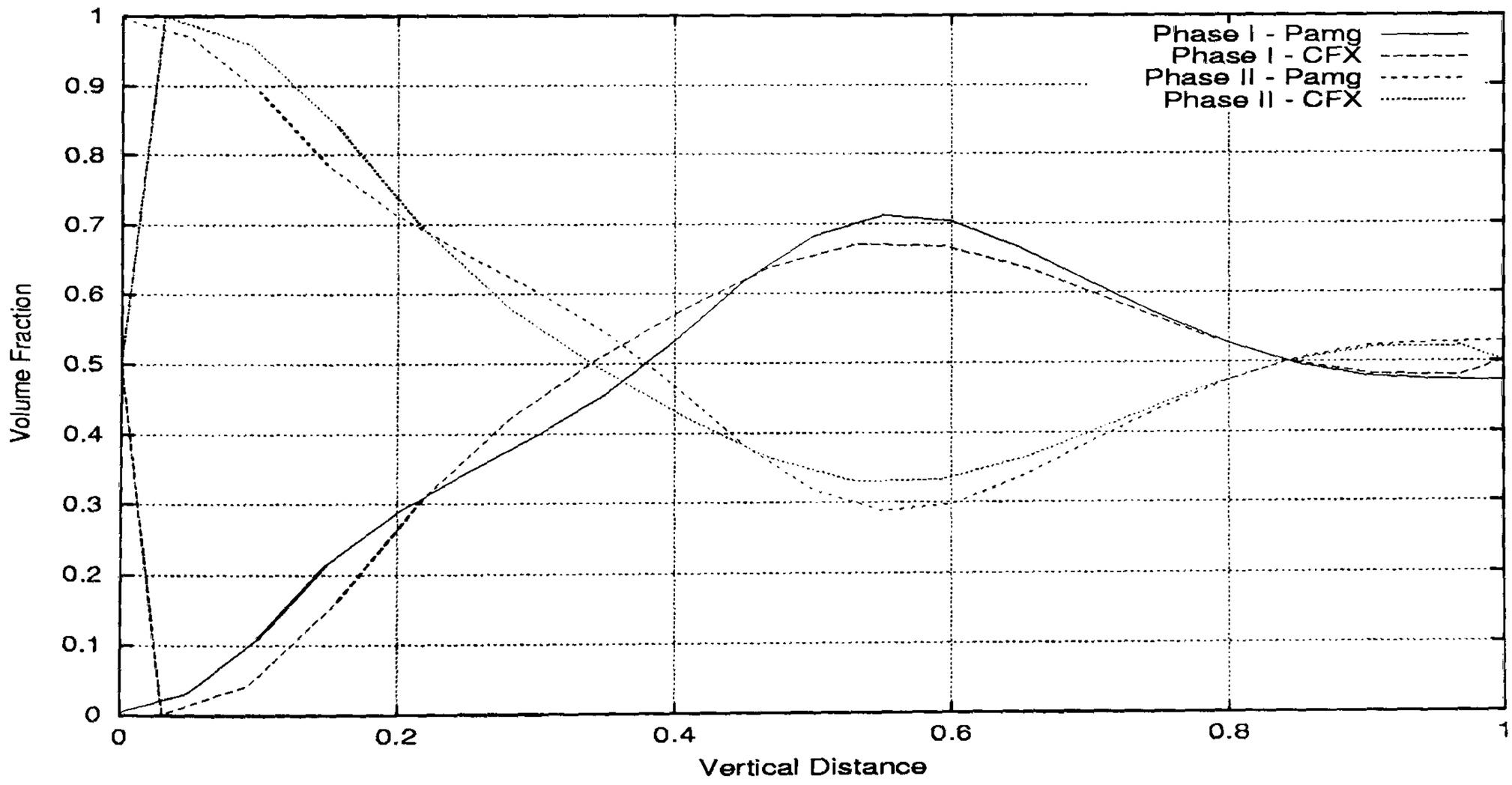


Figure 5.142: Multi-phase T-junction flow - comparison of volume fraction along the line $x = 3.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen the different initial volume fractions in the wall used by both solvers.

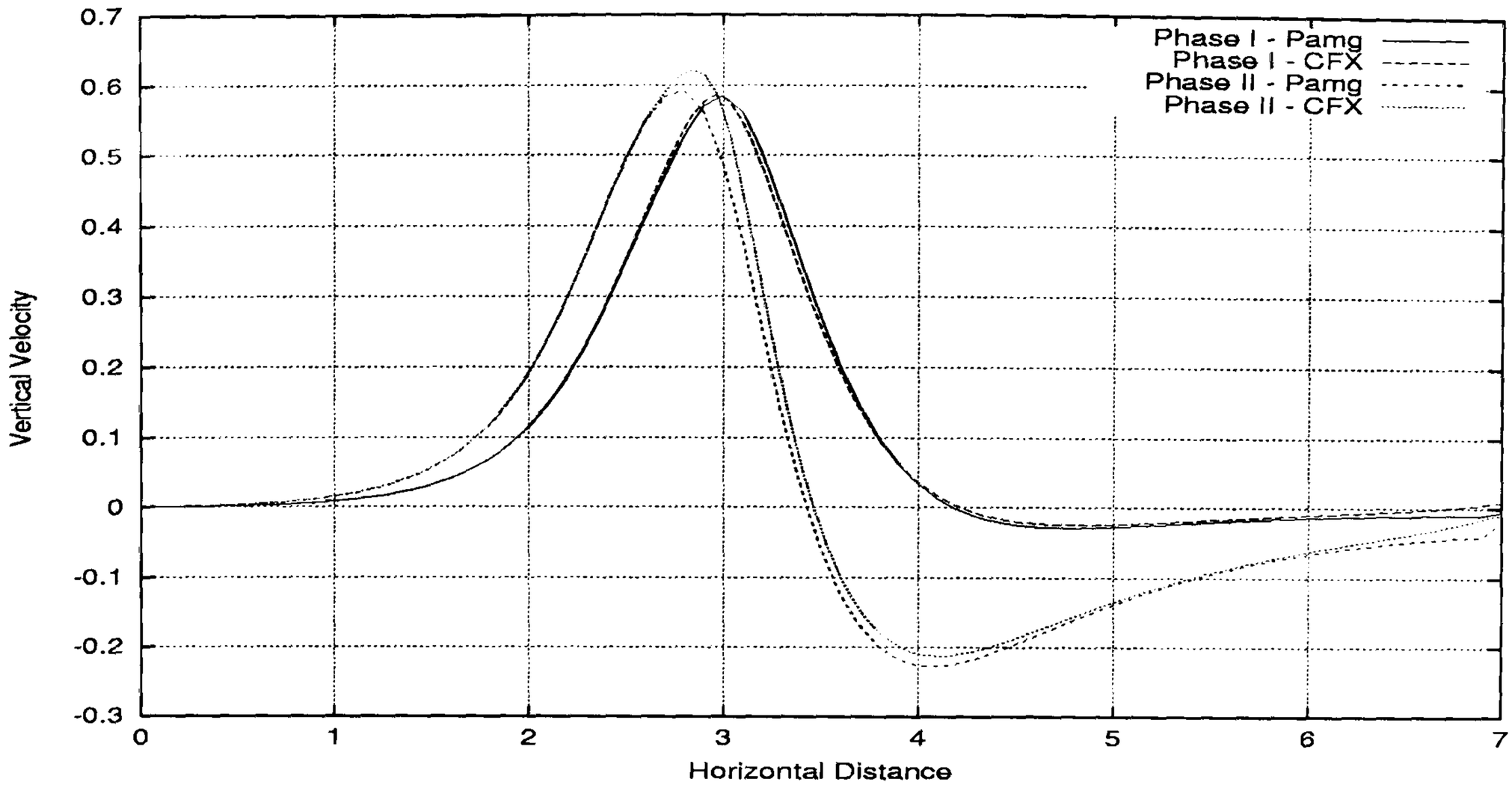


Figure 5.143: Multi-phase T-junction flow - comparison of vertical velocity along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen good agreement between the vertical velocities produced by both solvers.

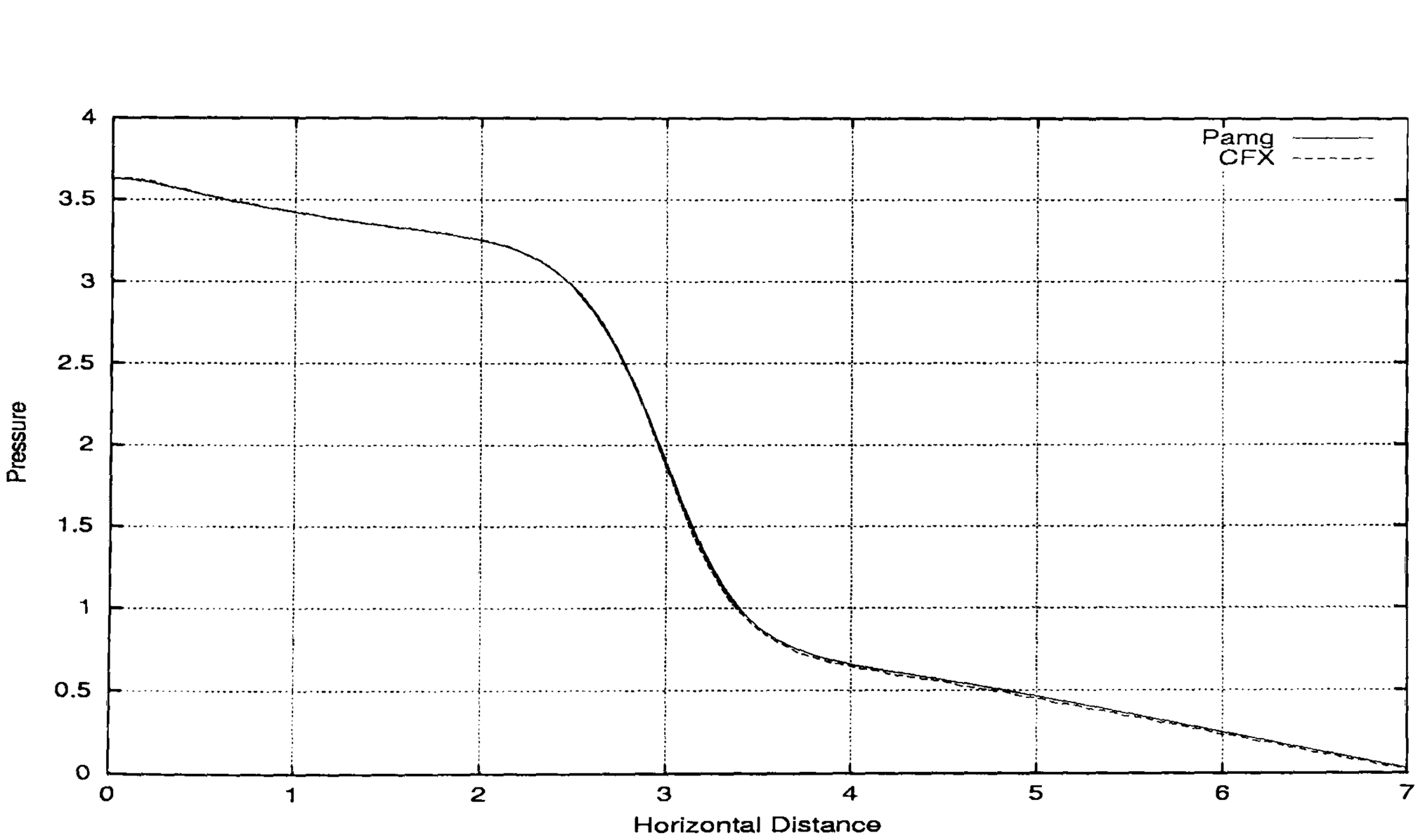


Figure 5.144: Multi-phase T-junction flow - comparison of pressure drop profiles along the line $y = 0.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen excellent agreement between the pressure values computed by both solvers.

5.3.3 Two-Phase Flow Through a Contraction

Problem definition

The final test problem to validate the `transient Pamg Multi-phase` is the simulation of two fluids trough a contraction. The geometry of this test case is more complex than in the two other problems studied before. This is due to the geometric constraint in the contraction region. Consequently the horizontal velocities change very quickly from small to bigger values. Moreover they change from positive to negative values very quickly after the contraction region. This happens due to the recirculation of the fluid flows. Naturally the vertical velocities exhibit big variations in this region either. Consequently this test case is much more demanding for the solution algorithm. Hence the main objective of this problem is to validate the code in a very complex geometry with a recirculation region. The physical properties of the two fluids are:

Two-Phase Flow	Reynolds Number	Viscosity	Density
Phase I	10	0.3	1.0
Phase II	7.5	0.2	0.5

In this test case, the values chosen for the drag coefficient and the inter-facial length are as in the T-junction:

$$C_D = 1.0$$

$$d_{\alpha\beta} = d_{\beta\alpha} = 0.1$$

The domain of this test case is represented in Figure 5.145. The computational grid at the uniform level 3 is presented in Figure 5.146. The resolution of the coarsest grid is $\Delta_{xsize} = \Delta_{ysize} = 0.25$. In Figure 5.147 we present the sections where the solution profiles are going to be presented. The boundary conditions are settled as follows. In the inlet the horizontal velocity profile is defined for each phase as $u(0, y) = 0.4y(3 - y)$ and the vertical velocity profile as $v(0, y) = 0$. The volume fractions are set to 0.5 for each phase. In the outlet the Neumann conditions are established. The initial condition is exactly the same for the T-junction problem. The velocities are all set to zero and the volume fractions to 0.5. The convergence tolerance for the simulation is 10^{-6} for each time step.

Results for a 3 level computation

In Figures 5.148 and 5.149 we see the streamlines for the two phases after 20 time units. It is visible in Figure 5.149 the recirculation region for *Phase II* just after the contraction.

Figures 5.150 through 5.159 present the profiles of the horizontal velocity, vertical velocity, volume fraction and pressure drop along the lines $x = 4.5$, $x = 9.0$(outlet) and $y = 1.5$ for 3 level computation. In Figure 5.150 we present the horizontal velocity along the line $y = 1.5$ for a 3 level computation. The velocity

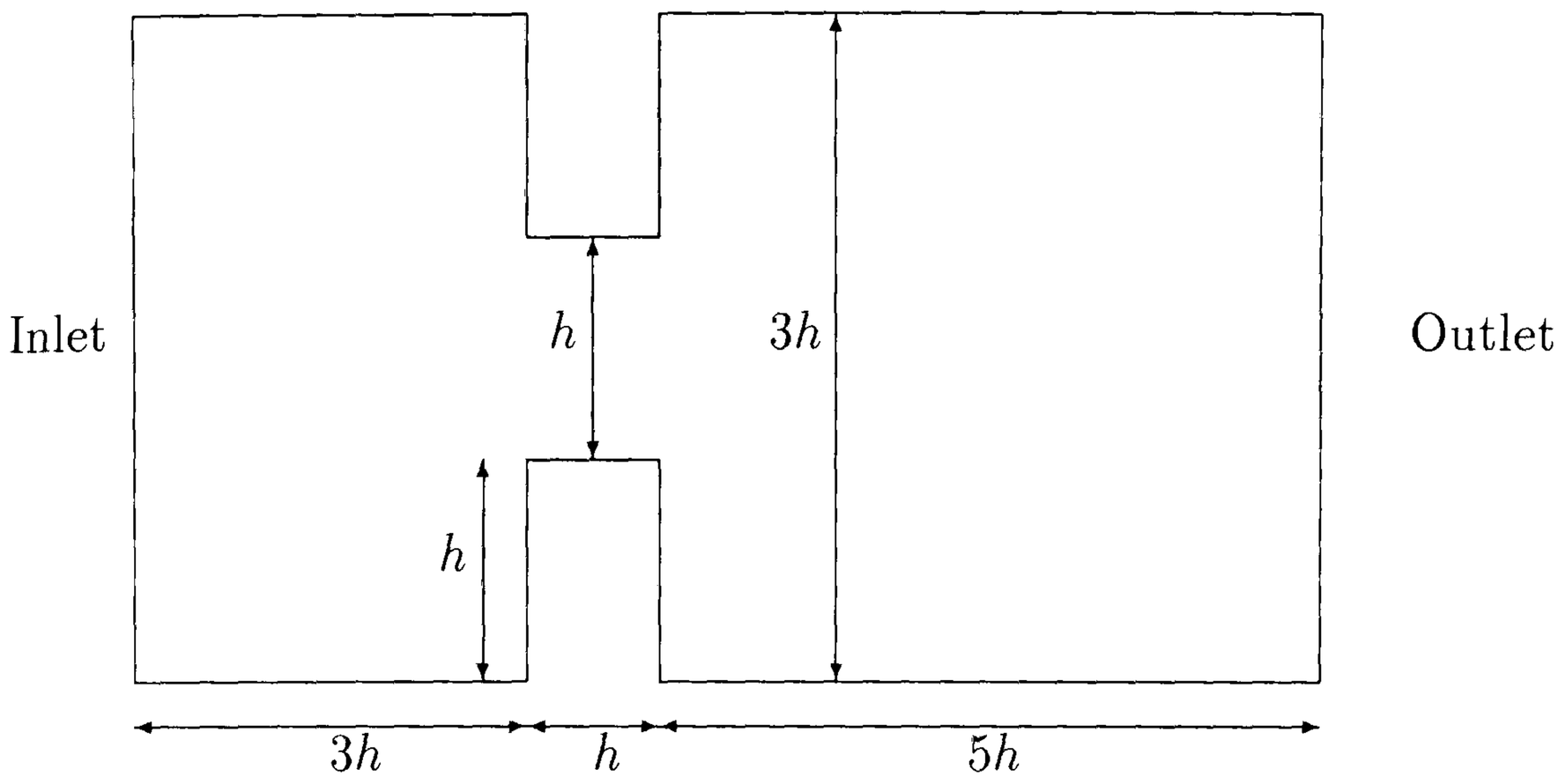


Figure 5.145: Geometrical representation of the domain of the Contraction two-phase flow problem

grows up in the contraction region and afterwards decreases until reaching the velocity of the initial guess. The velocities at 20 time units are naturally bigger than at 1 time unit.

Figure 5.151 presents the volume fraction profile along the line $y = 1.5$ for a 3 level computation. It can be seen that after 20 time units the volume fractions of both phases are much more close. Furthermore there exist two points where the volume fractions are identical.

Figure 5.152 presents the horizontal velocity along the line $x = 4.5$ for 3 level computation. It is visible the parabolic profile presented by both curves which is more evident at 20 time units.

In Figure 5.153 it can be seen the vertical velocity along the line $x = 4.5$ for 3 level computation. They are negative until the middle of the vertical distance and positive afterwards.

Figure 5.154 shows the volume fraction along the line $x = 4.5$ for a 3 level computation.

Figure 5.155 presents the pressure drop along the line $y = 1.5$ for a 3 level computation. It is visible that the pressure changes linearly in all the regions of the domain. In the contraction region the variation is bigger which is precisely what it should be expected.

Figure 5.156 presents the vertical velocity along the line $y = 1.5$ for a 3 level computation. It is visible the variation between negative and positive values in the contraction region.

In Figure 5.157 we present the Horizontal Velocity along the line $x = 9.0$ for a 3 level computation. It can be seen the parabolic profile which is more evident at 20 time units.

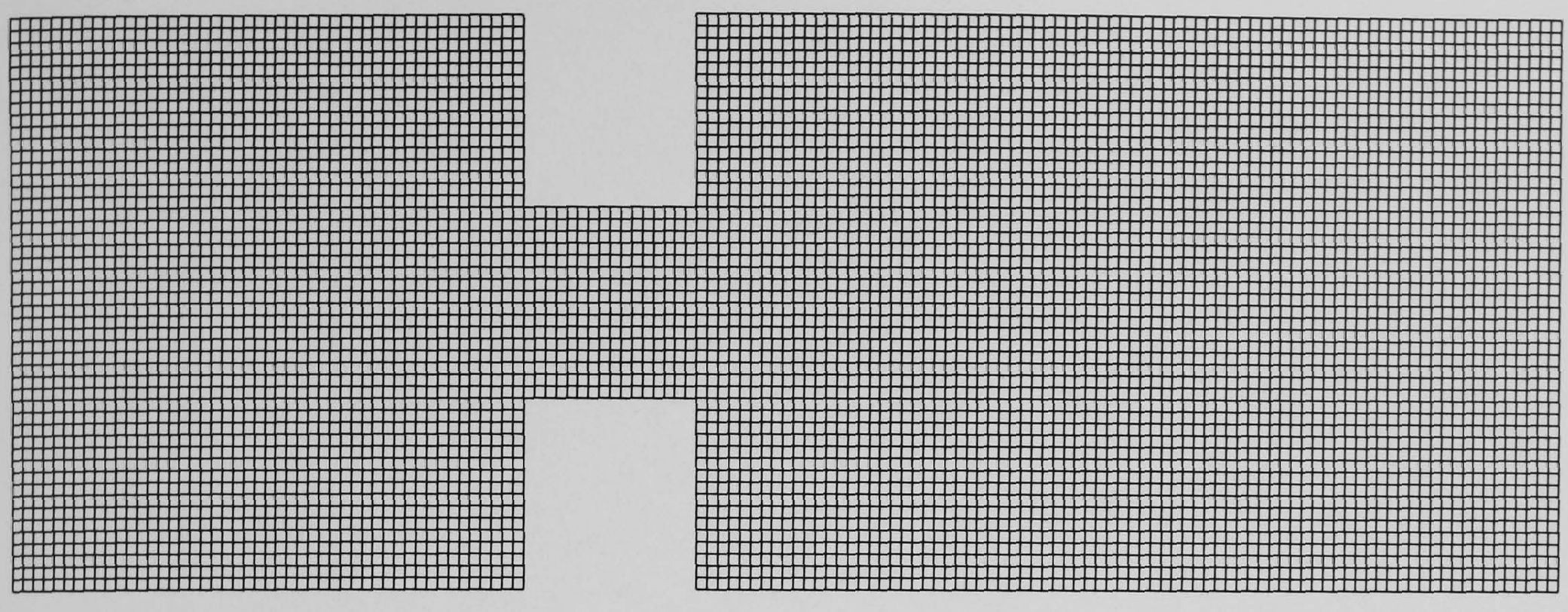


Figure 5.146: Uniform level 3 grid for the Contraction two-phase flow test case ($\Delta_{xsize} = 0.0625$ and $\Delta_{ysize} = 0.0625$)

Figure 5.158 shows the Vertical Velocity along the line $x = 9.0$ for a 3 level computation. Since we are pre-setting this velocity in the outlet ($x = 9.0$) the values are very small and even zero.

In Figure 5.159 we present the volume fraction along the line $x = 9.0$ for a 3 level computation.

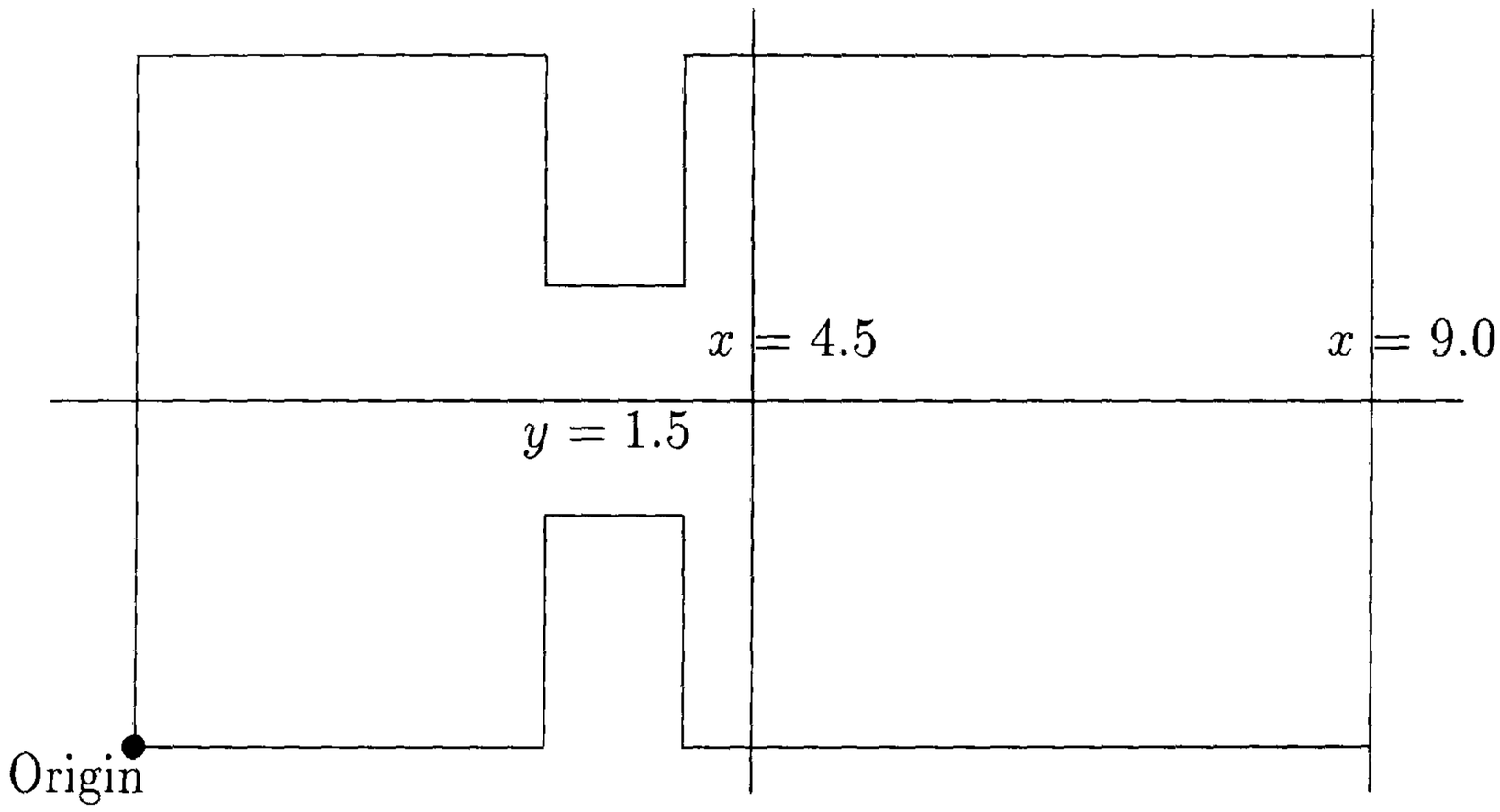


Figure 5.147: Contraction two-phase flow test case - main sections used to give the solution profiles

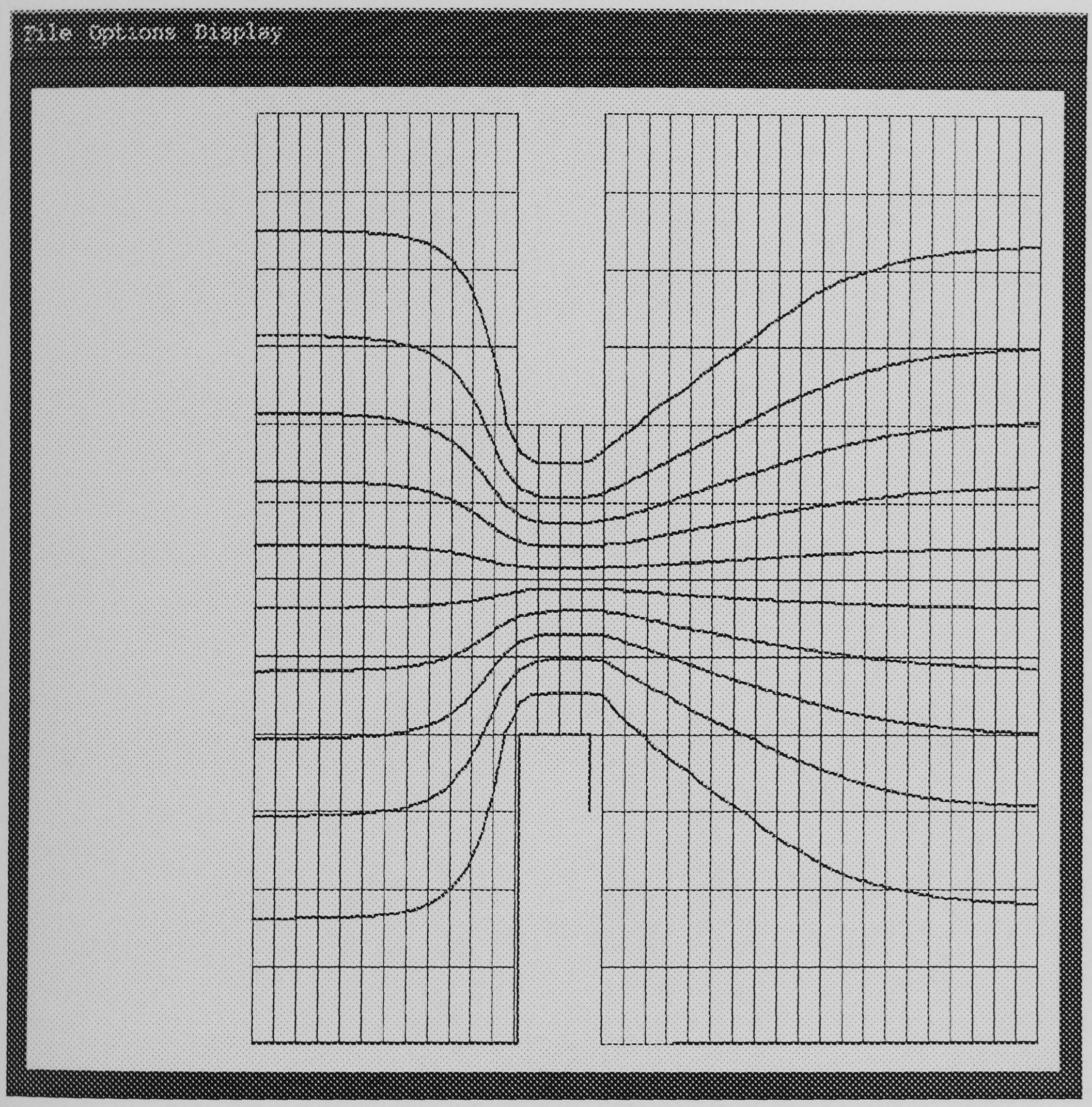


Figure 5.148: Multi-phase Contraction flow - streamlines after 20 time units for *Phase I*

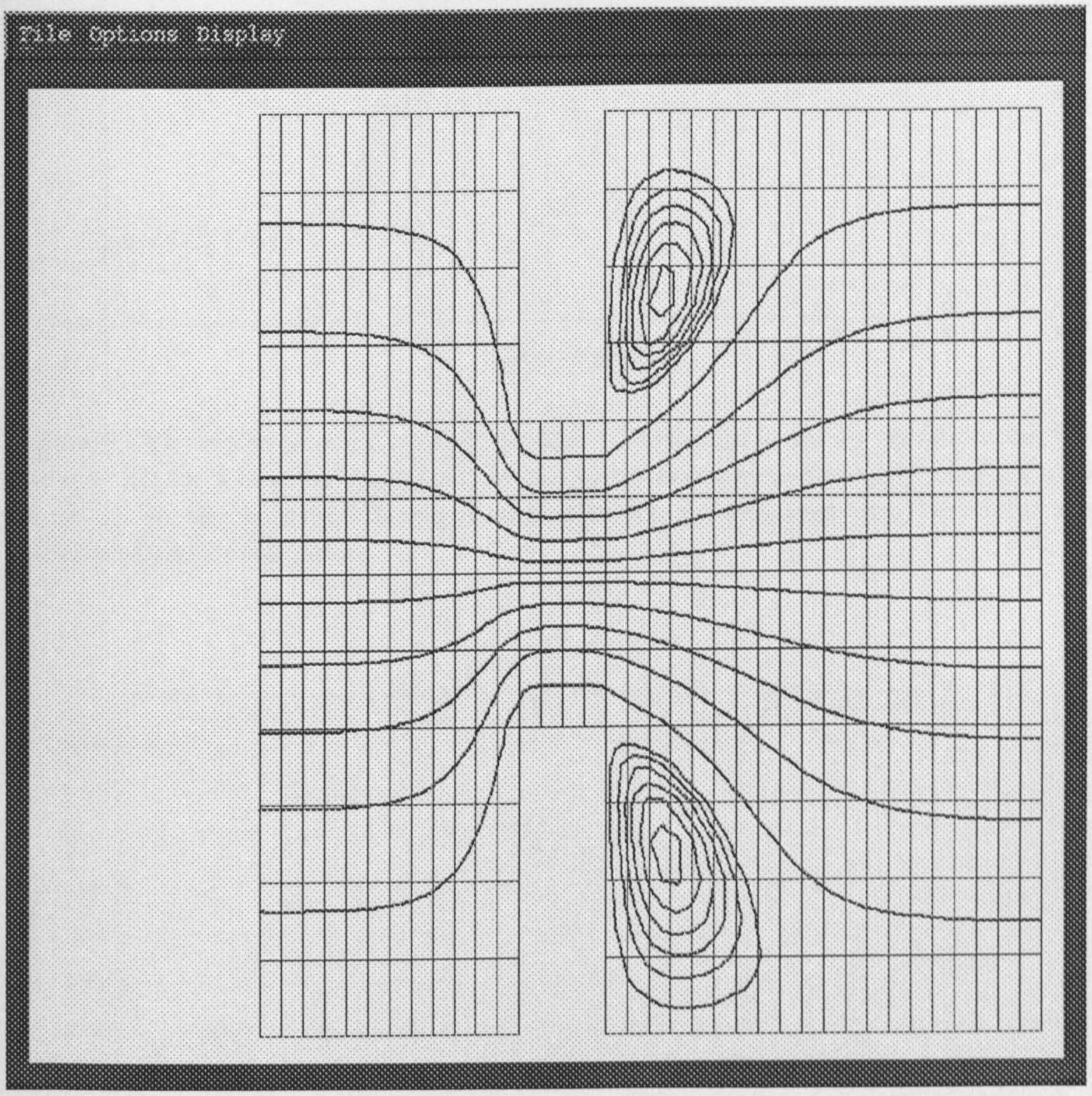


Figure 5.149: Multi-phase Contraction flow - streamlines after 20 time units for *Phase II*

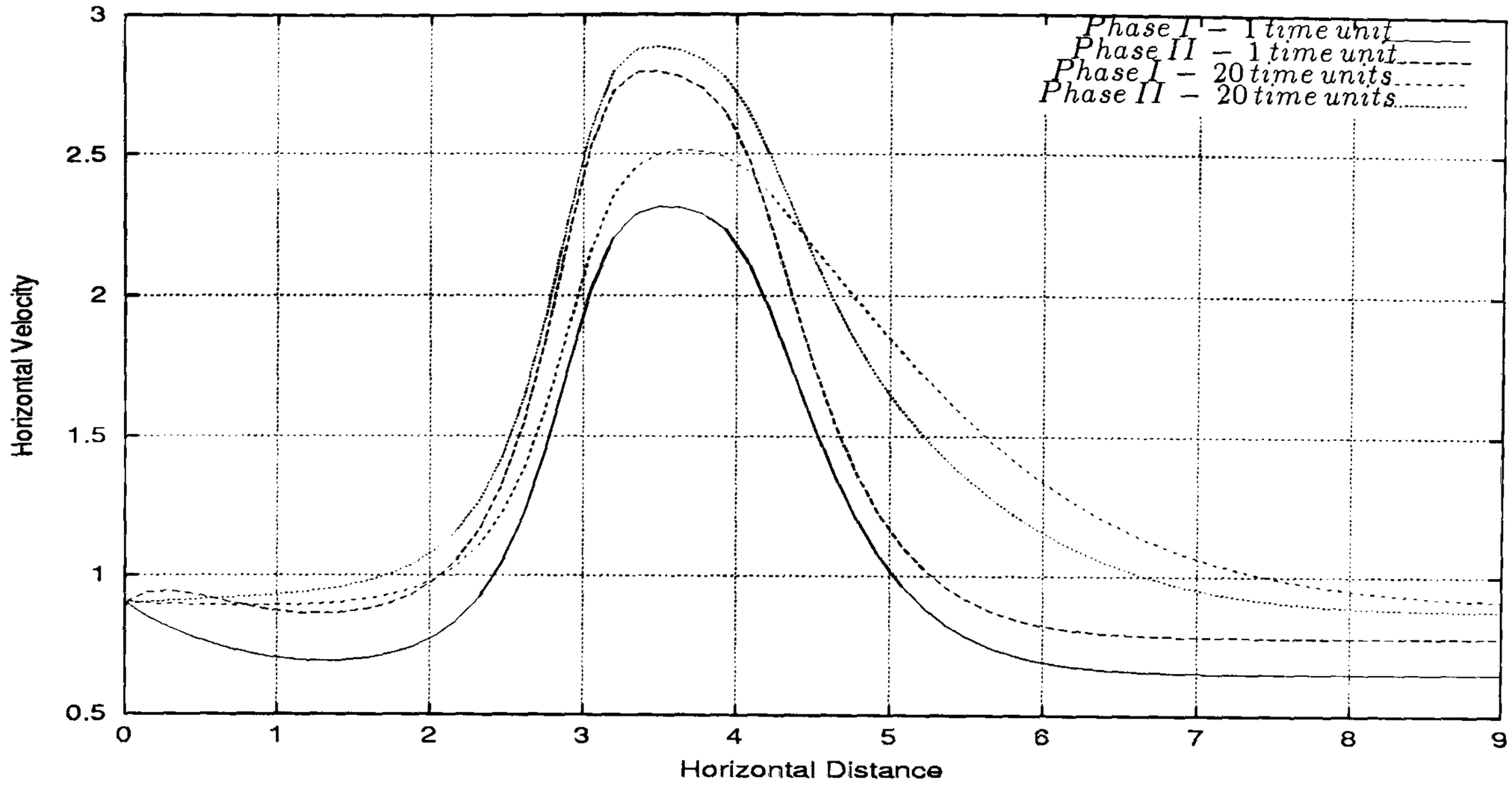


Figure 5.150: Multi-phase Contraction flow - horizontal velocity profile along the line $y = 1.5$ for 3 level computation. It can be observed that the horizontal velocity grows up in the contraction region and afterwards decreases until reaching the velocity of the initial guess.

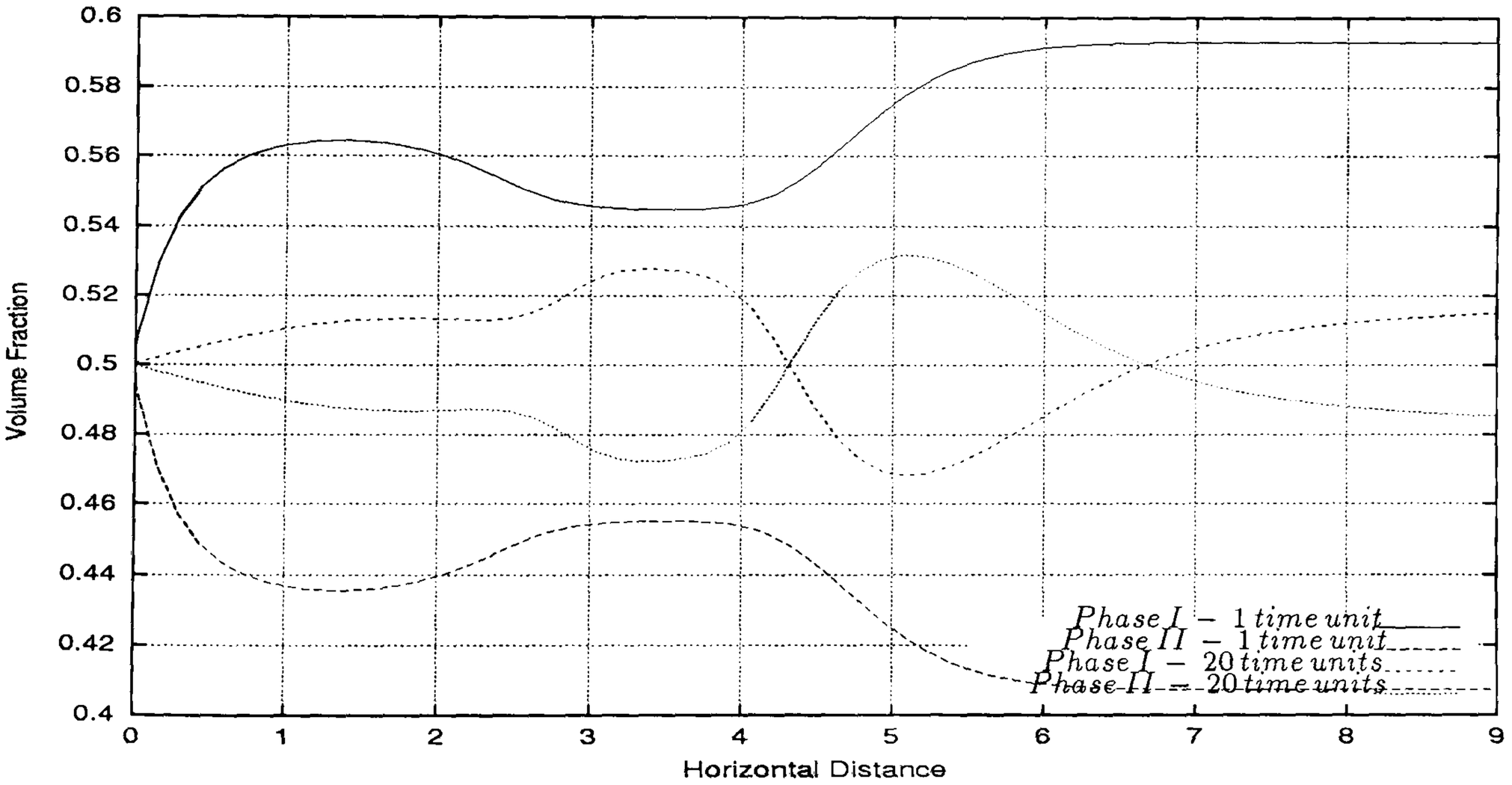


Figure 5.151: Multi-phase Contraction flow - volume fraction profile along the line $y = 1.5$ for 3 level computation. It can be seen that after 20 time units the volume fractions of both phases are closer than at 1 time unit. Note the two points where the volume fractions are identical.

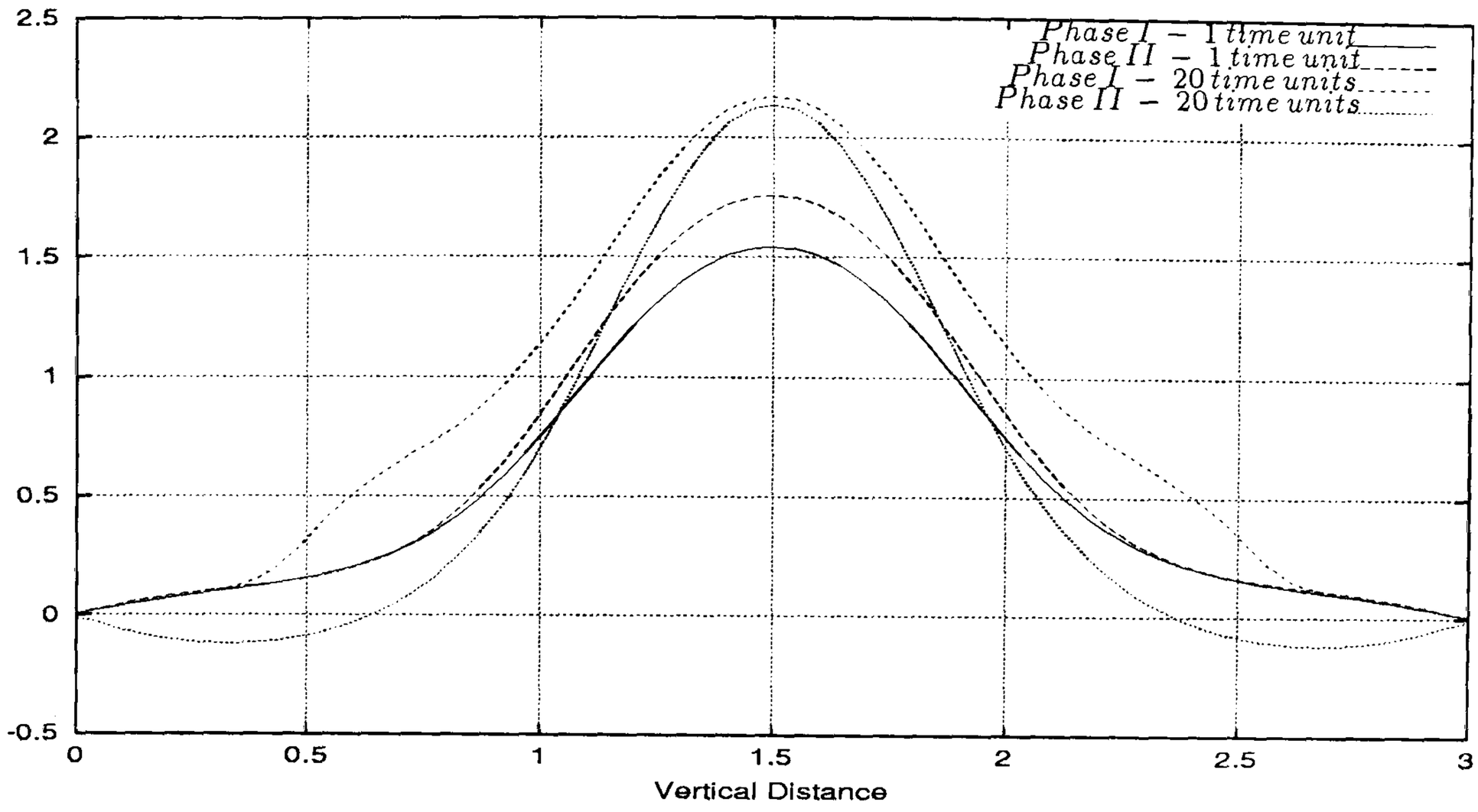


Figure 5.152: Multi-phase Contraction flow - horizontal velocity profile along the line $x = 4.5$ for 3 level computation. It can be observed, particularly at 20 time units, the parabolic profile of the velocities for both phases.

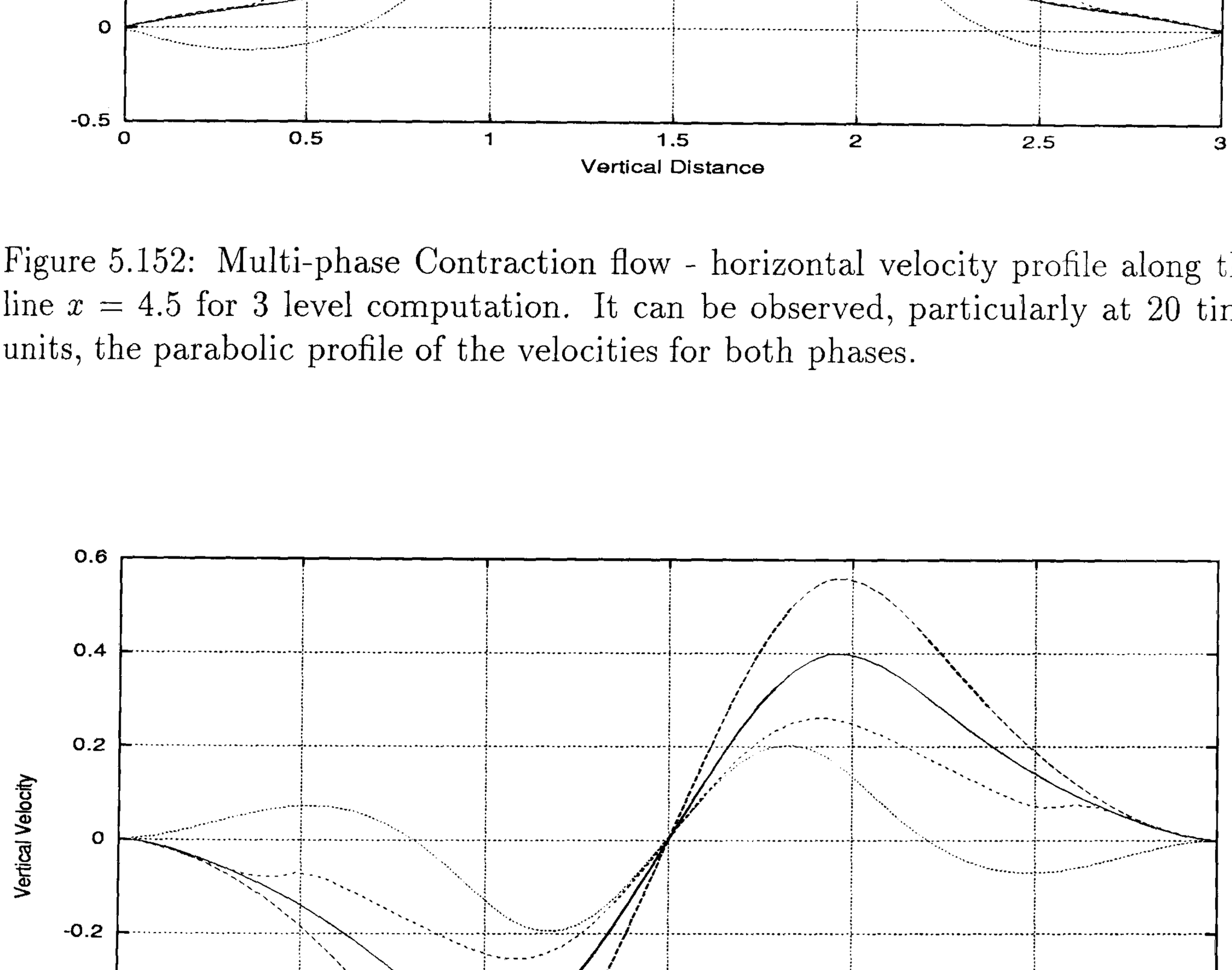


Figure 5.153: Multi-phase Contraction flow - vertical velocity profile along the line $x = 4.5$ for 3 level computation. It can be seen that the vertical velocity is negative until the middle of the vertical distance and positive afterwards.

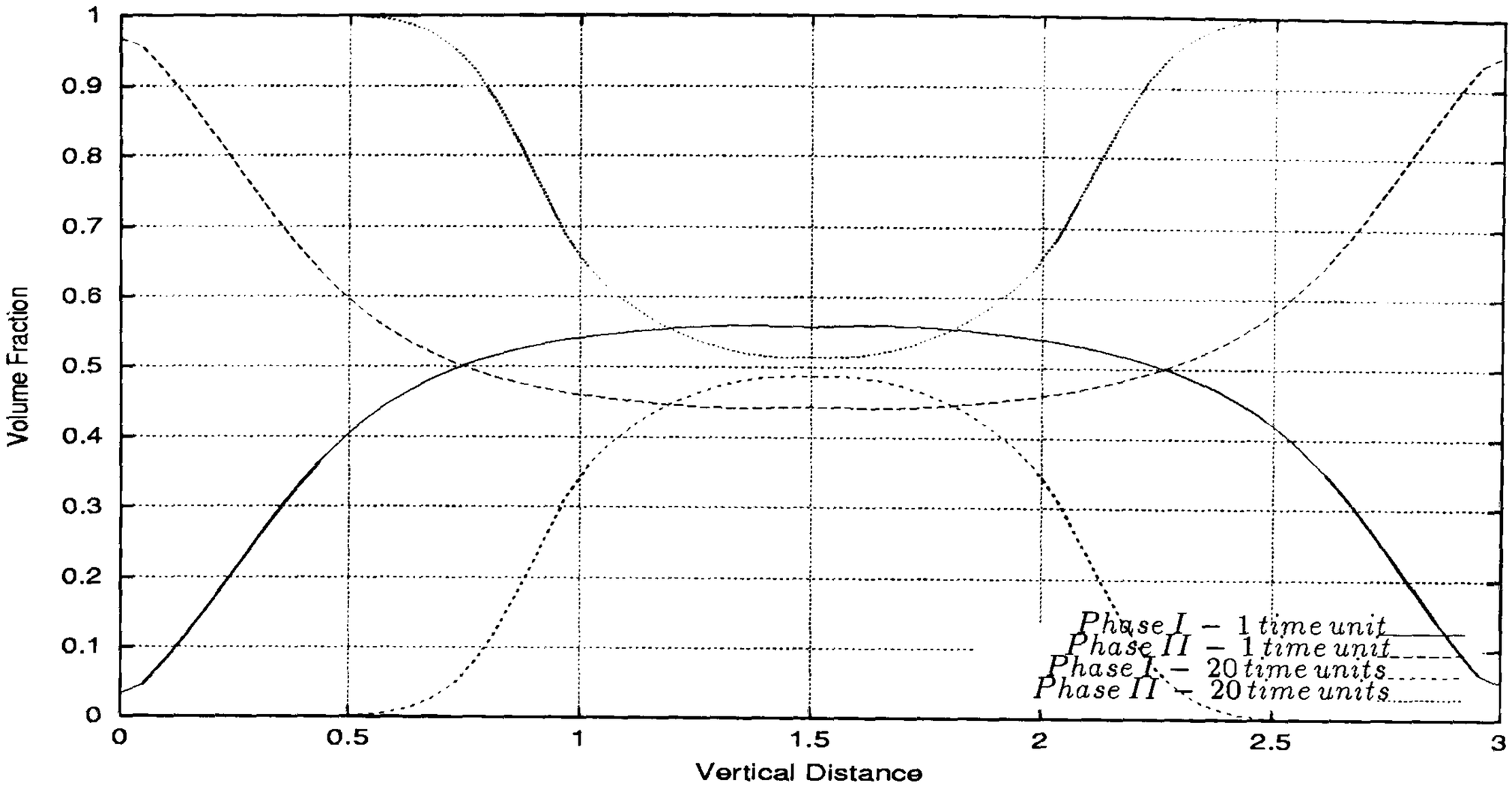


Figure 5.154: Multi-phase Contraction flow - volume fraction profile along the line $x = 4.5$ for 3 level computation. It can be seen that the volume fraction approach each other in the middle of the vertical distance at 20 time units.

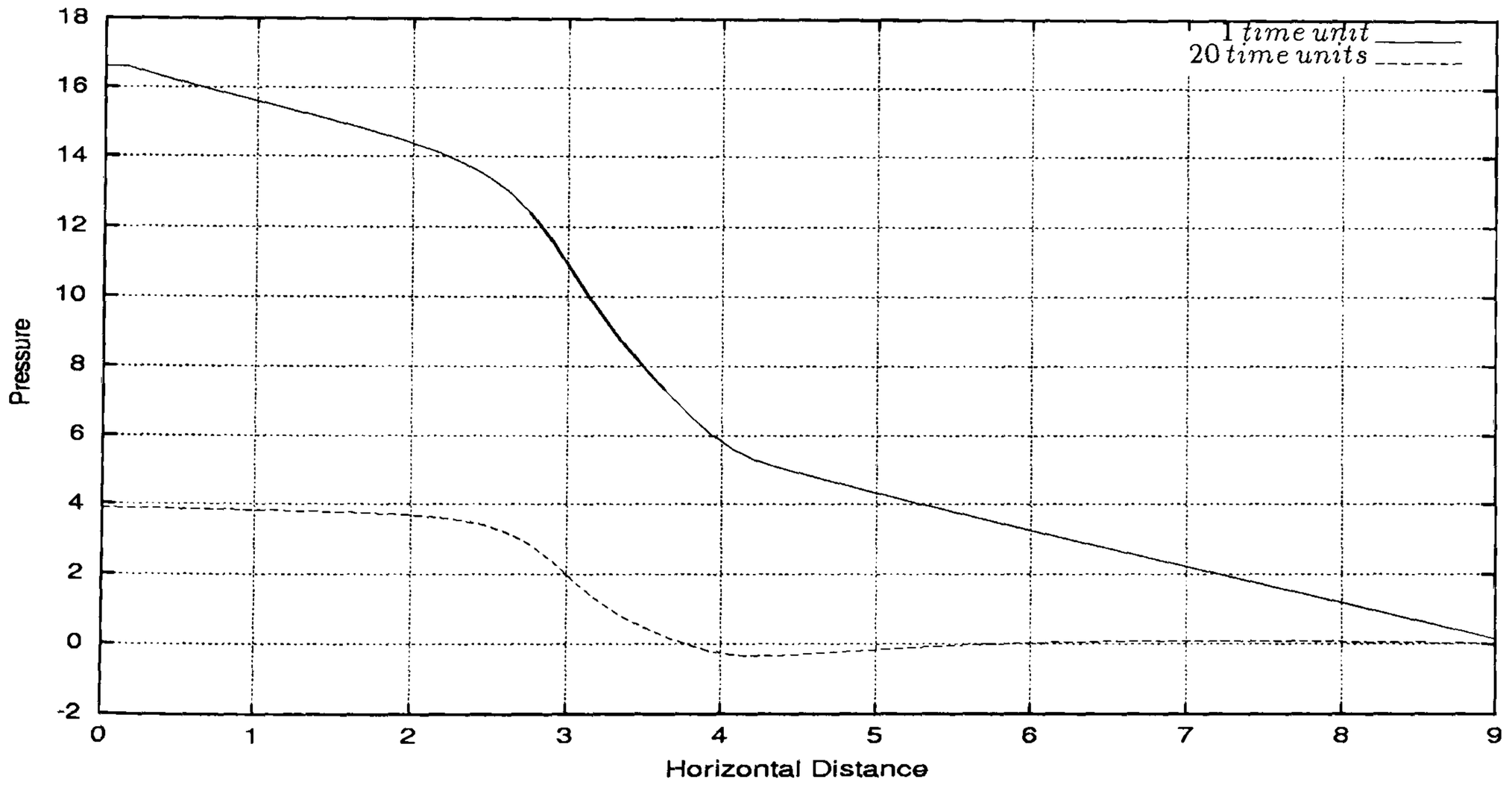


Figure 5.155: Multi-phase Contraction flow - pressure profile along the line $y = 1.5$ for 3 level computation. It can be observed the bigger pressure variation in the contraction region both at 1 and 20 time units.

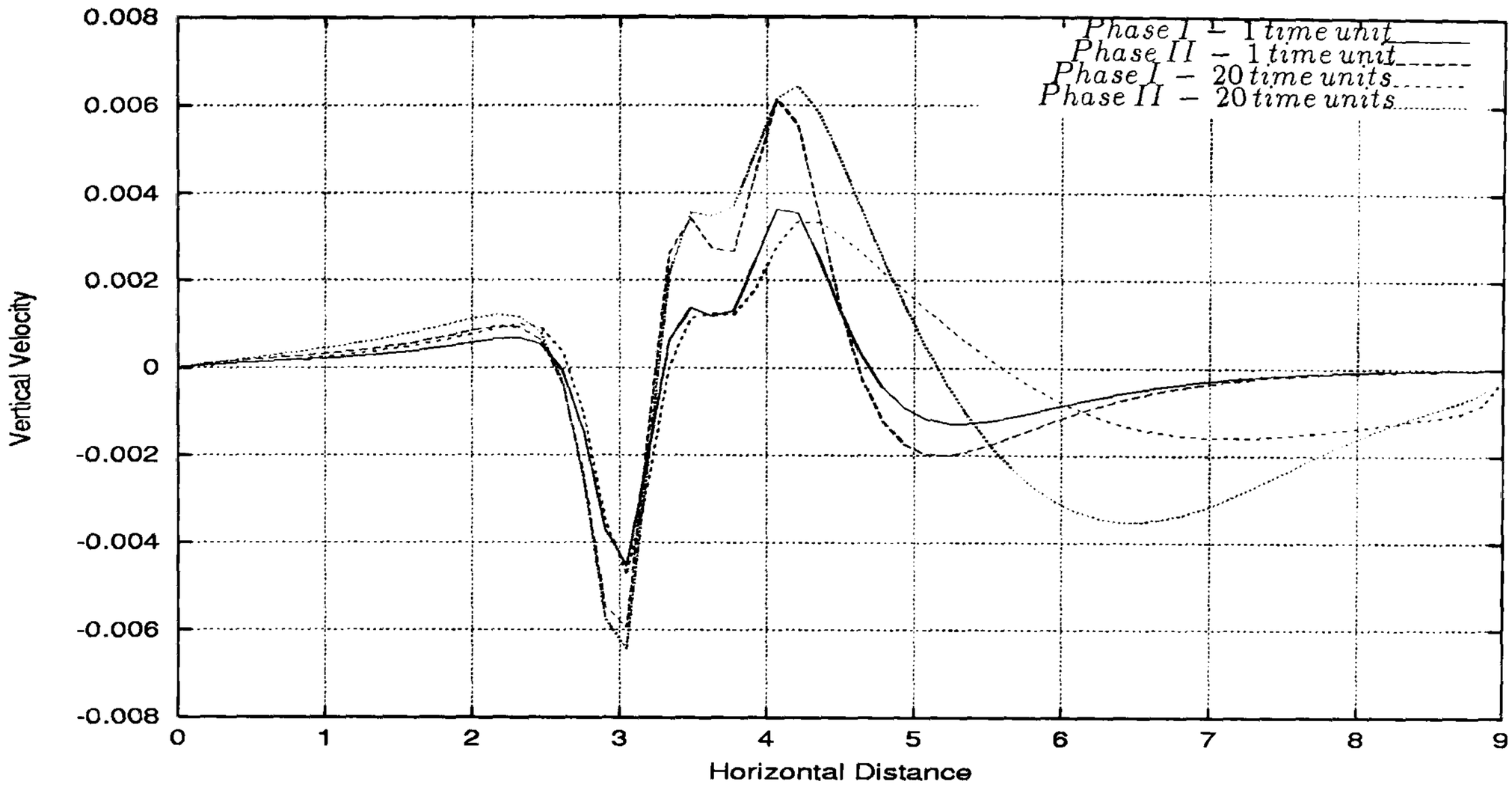


Figure 5.156: Multi-phase Contraction flow - vertical velocity profile along the line $y = 1.5$ for 3 level computation. It can be seen the variation between negative and positive values in the contraction region.

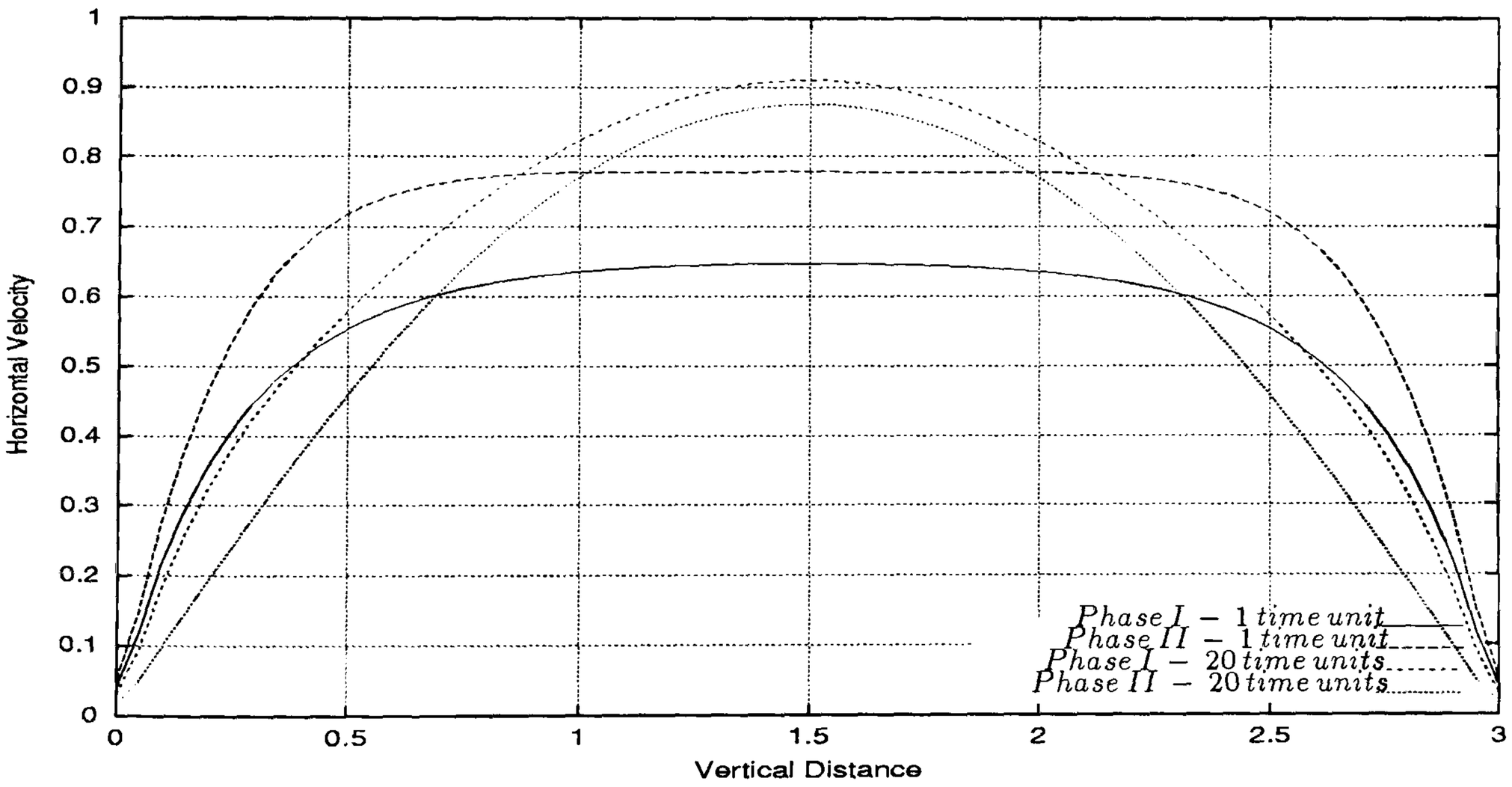


Figure 5.157: Multi-phase Contraction flow - horizontal velocity profile along the line $x = 9.0$ for 3 level computation. It can be seen the fully developed parabolic profile at 20 time units.

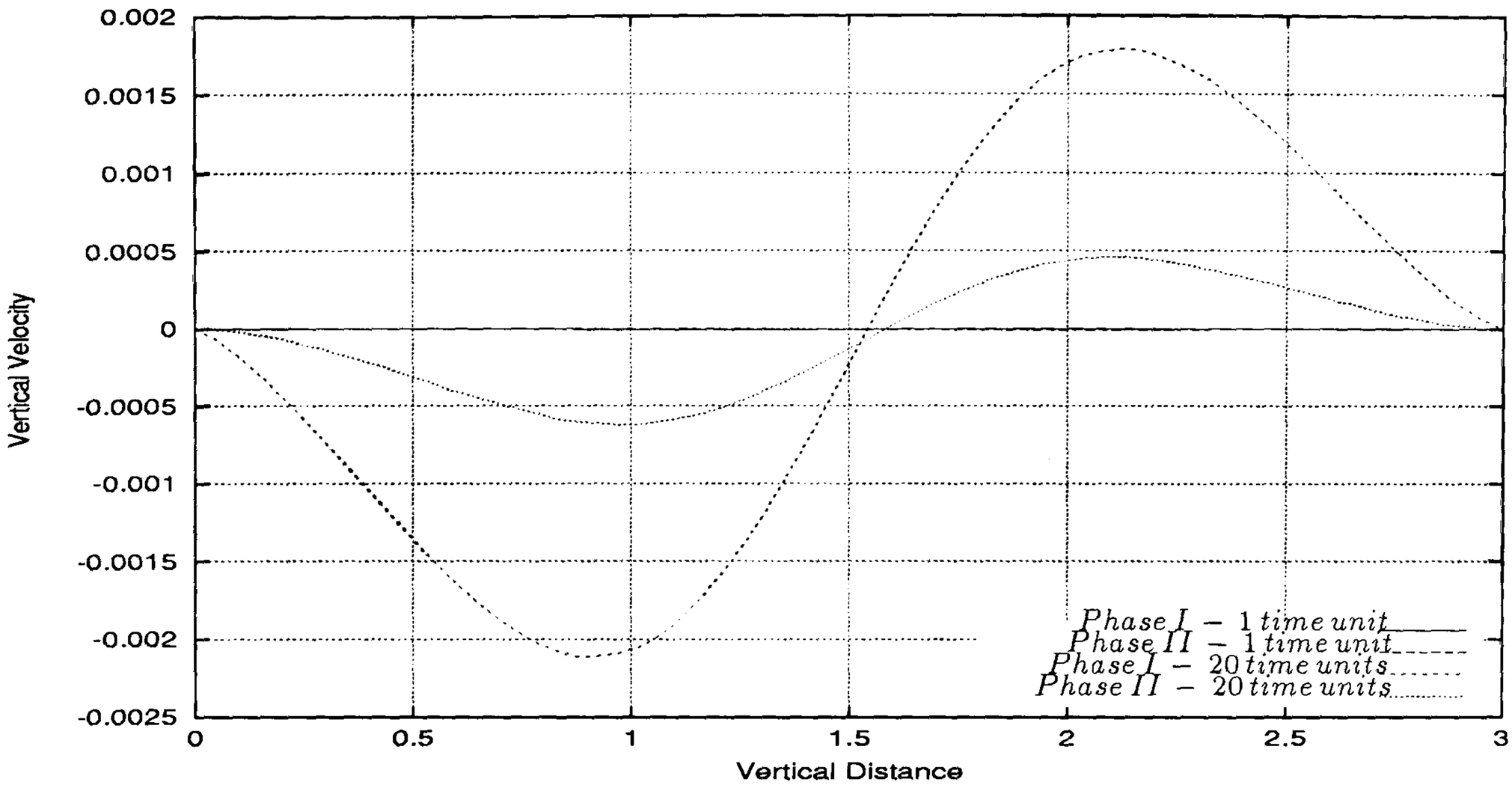


Figure 5.158: Multi-phase Contraction flow - vertical velocity profile along the line $x = 9.0$ for 3 level computation. It can be seen that the velocity values are very small.

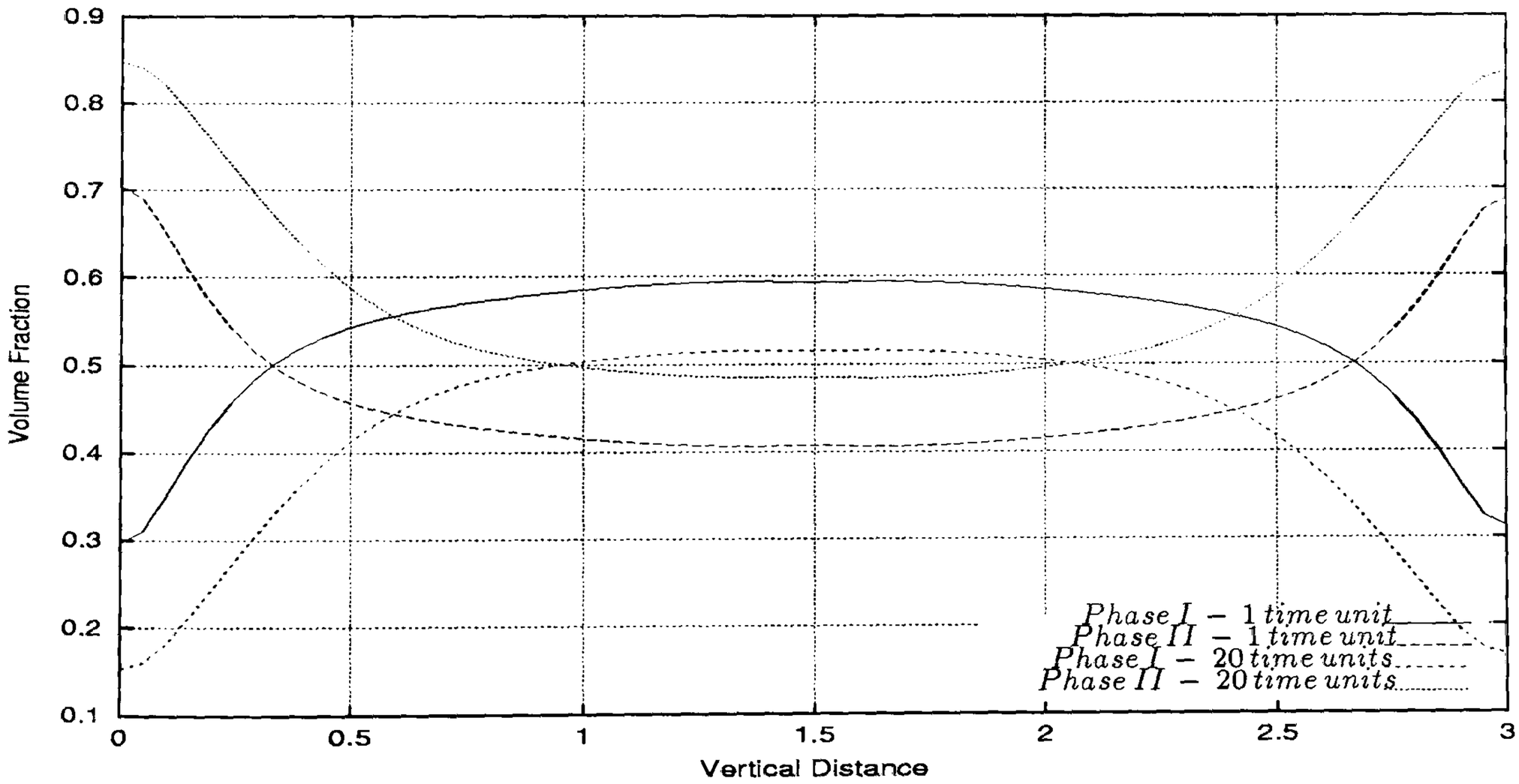


Figure 5.159: Multi-phase Contraction flow - volume fraction profile along the line $x = 9.0$ for 3 level computation. It can be seen that at 20 time units the volume fractions approach the value 0.5 in the middle of the vertical distance.

Grid Independence of the Solutions

Figures 5.160 through 5.178 present the profiles of the horizontal velocity, vertical velocity, volume fraction and pressure drop along the lines $x = 4.5$, $x = 9.0$(outlet) and $y = 1.5$ for single 2 and 3 level computation.

In Figures 5.160 and 5.161 we present the Horizontal Velocity for *Phase I* and *Phase II* along the line $y = 1.5$ for single 2 and 3 level computations. It is visible that in the contraction region the agreement between the single 2 and 3 grids curves is not so good as in the initial and final region. This must be due to the geometric constraint that originates big changes in the variation of the horizontal velocities.

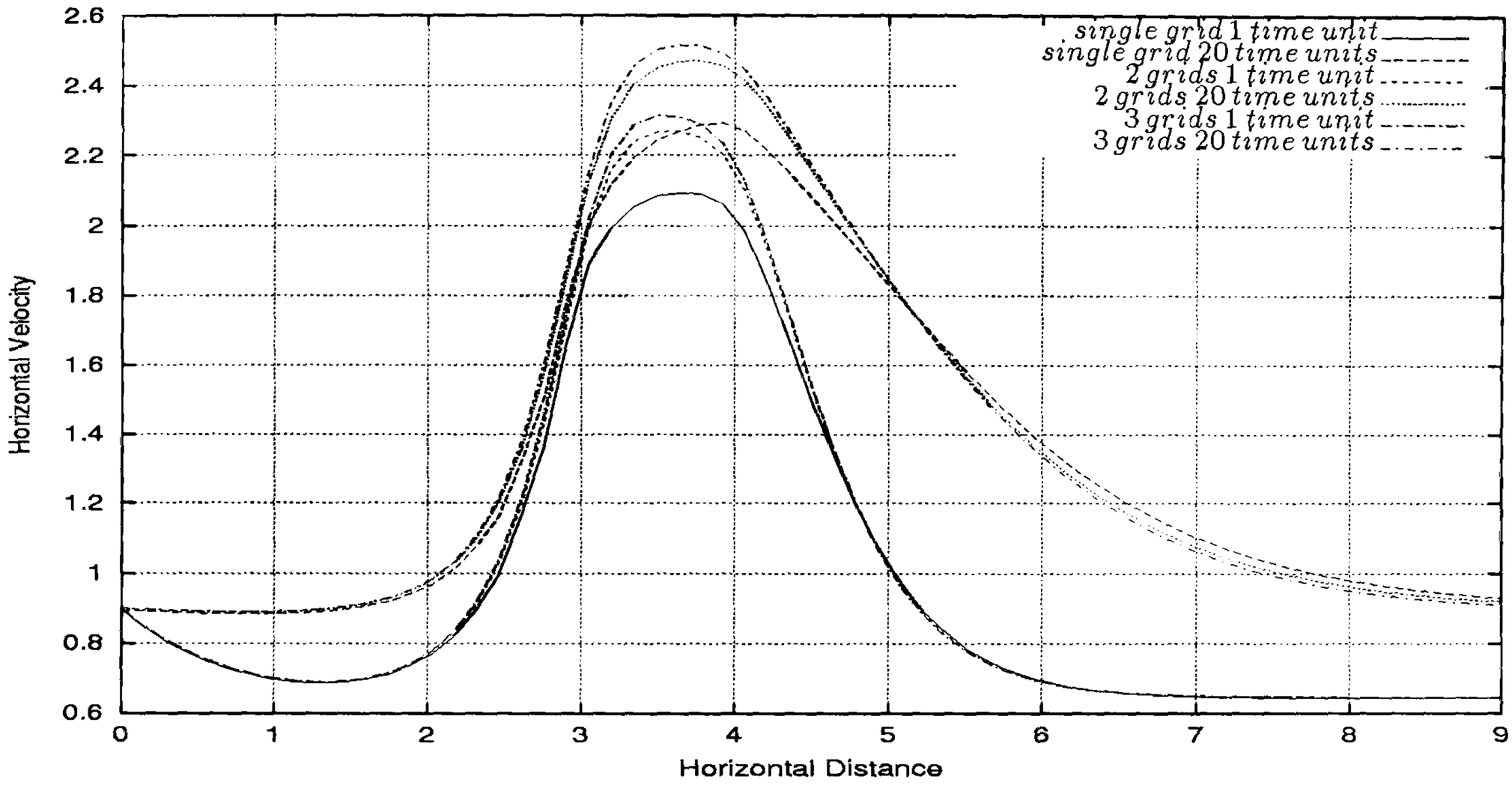


Figure 5.160: Multi-phase Contraction flow - horizontal velocity profile for *Phase I* along the line $y = 1.5$ for single 2 and 3 level computation. It can be seen that in the contraction region the agreement between all the level computations is not so efective than the agreement observed in the initial and final regions of the domain.

Figures 5.162 and 5.163 present the volume fractions for *Phase I* and *Phase II* along the line $y = 1.5$ for single 2 and 3 level computation. While the volume fraction of *Phase I* increases the volume fraction of *Phase II* decreases. The agreement between the single 2 and 3 level computations is very good particularly at 1 time unit. Although the curves corresponding to 20 time units show some disagreement at the outlet we must remember that the scale of the graph is very small.

Figures 5.164 and 5.165 present the horizontal velocity for *Phase I* and *Phase II* along the line $x = 4.5$ for single 2 and 3 level computation. It is visible the parabolic profile and a very good agreement between the curves corresponding to all the 3 levels of computations.

In Figures 5.166 and 5.167 we show the vertical velocity along the line $x = 4.5$ for single, 2 and 3 level computation. The vertical velocity of both phases starts

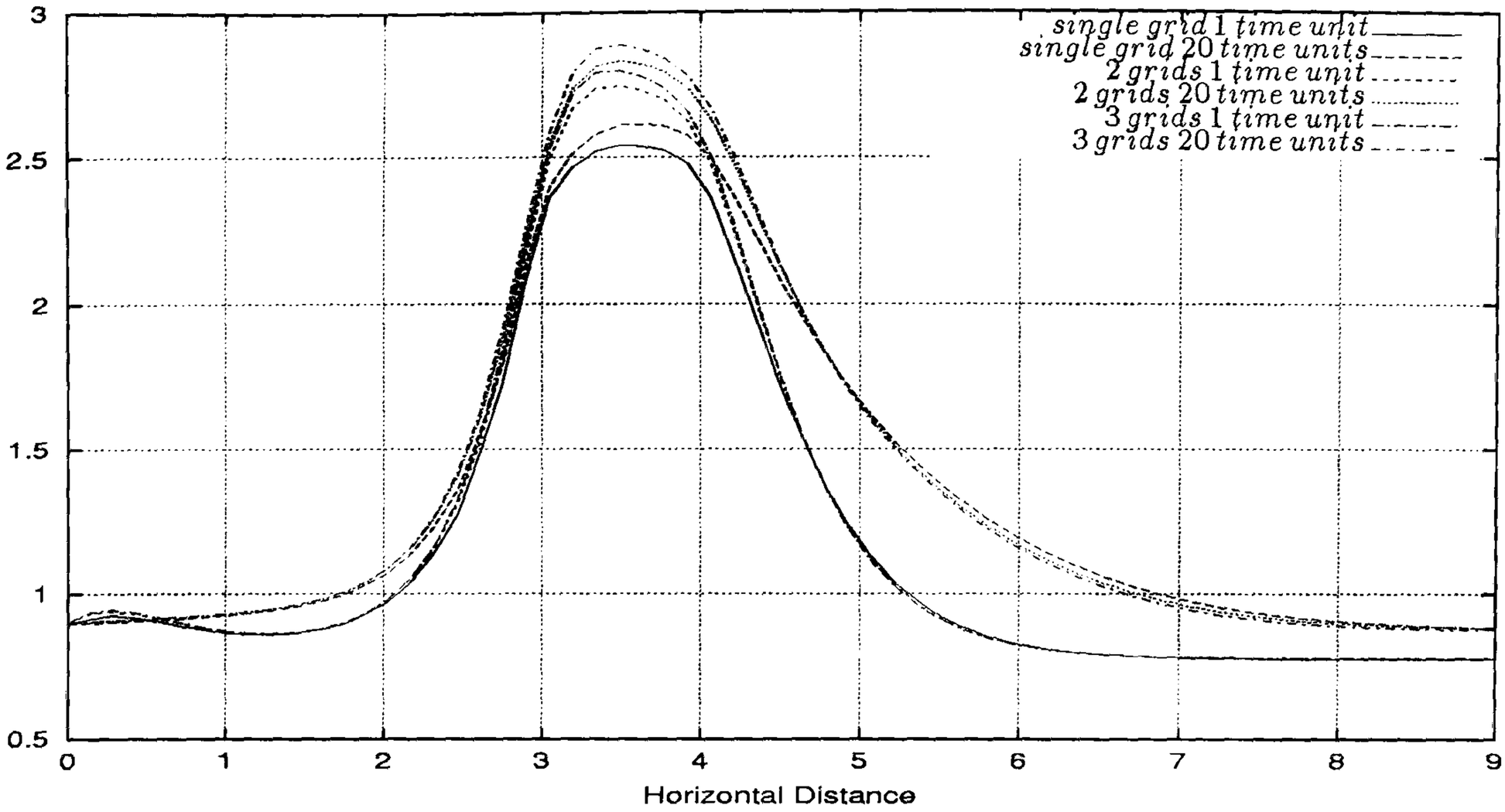


Figure 5.161: Multi-phase Contraction flow - horizontal velocity profile for *Phase II* along the line $y = 1.5$ for single 2 and 3 level computation. It can be seen that in the contraction region the agreement between all the level computations is not so efective than the agreement observed in the initial and final regions of the domain

and ends at zero varying from negative to positive values along all the entire vertical distance.

Figures 5.168 and 5.169 show the volume fraction for *Phase I* and *Phase II* along the line $x = 4.5$ for single 2 and 3 level computation. At 1 time unit the volume fractions computed by all the 3 levels computation are very different in the walls. However at 20 time units the differences are very small. In the middle of the vertical distance it can be seen good agreement in the curves corresponding to the 3 levels computation.

Figure 5.170 presents the pressure drop along the line $y = 1.5$ for single 2 and 3 level computation. The profile of the 3 different computations exhibit a very good agreement both at 1 and 20 time units.

In Figures 5.171 and 5.172 we show the vertical velocity along the line $y = 1.5$ for single, 2 and 3 level computations. In the single grid computations the vertical velocities are null as it can be deduced by the horizontal lines in the figures. Although some slight differences are observed in the curves of the 2 and 3 level computations, particularly after the horizontal distance of 3.5, the values at the outlet are very close to zero.

Figures 5.173 and 5.174 present the horizontal velocity for *Phase I* and *Phase II* along the line $x = 9.0$. The fully developed parabolic profile can be observed at 20 time units for all the level computations. Moreover the agreement between all these curves is very good with small differences in the walls which are due to the interpolation errors.

In Figures 5.175 and 5.176 we present the vertical velocity for *Phase I* and

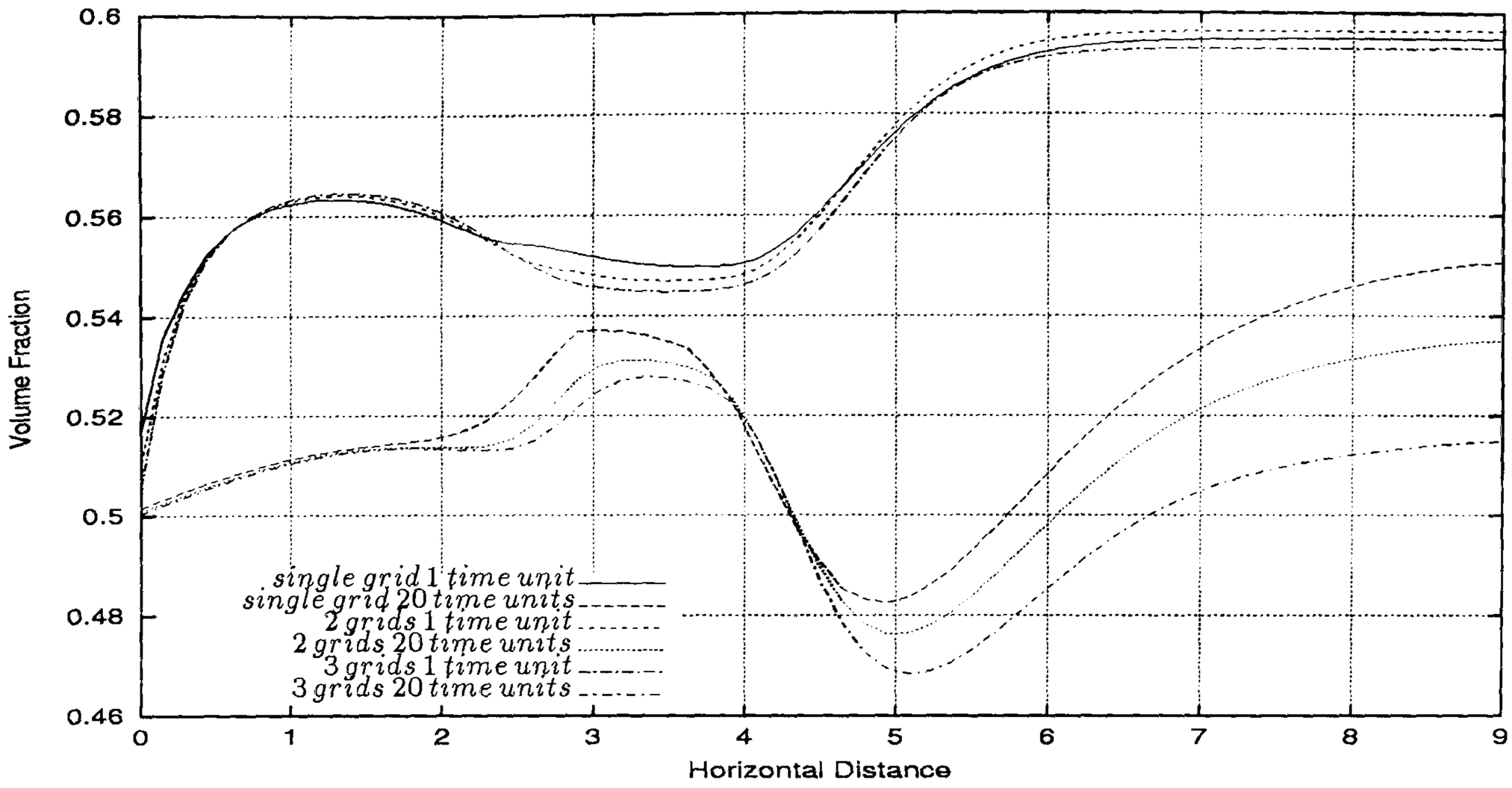


Figure 5.162: Multi-phase Contraction flow - volume fraction profile for *Phase I* along the line $y = 1.5$ for single 2 and 3 level computation. It can be seen good agreement between all the level computations. Note the small scale of the volume fraction axis.

Phase II along the line $x = 9.0$. Since we are in the outlet of the pipe the vertical velocities are very small and close to zero.

Finally in Figures 5.177 and 5.178 we show the volume fraction profiles for *Phase I* and *Phase II* along the lines $x = 9.0$ for single, 2 and 3 level computation. It is visible some disagreement in the walls between all the 3 level computations. However at 20 time units the differences between the 3 curves are smaller than at 1 time unit.

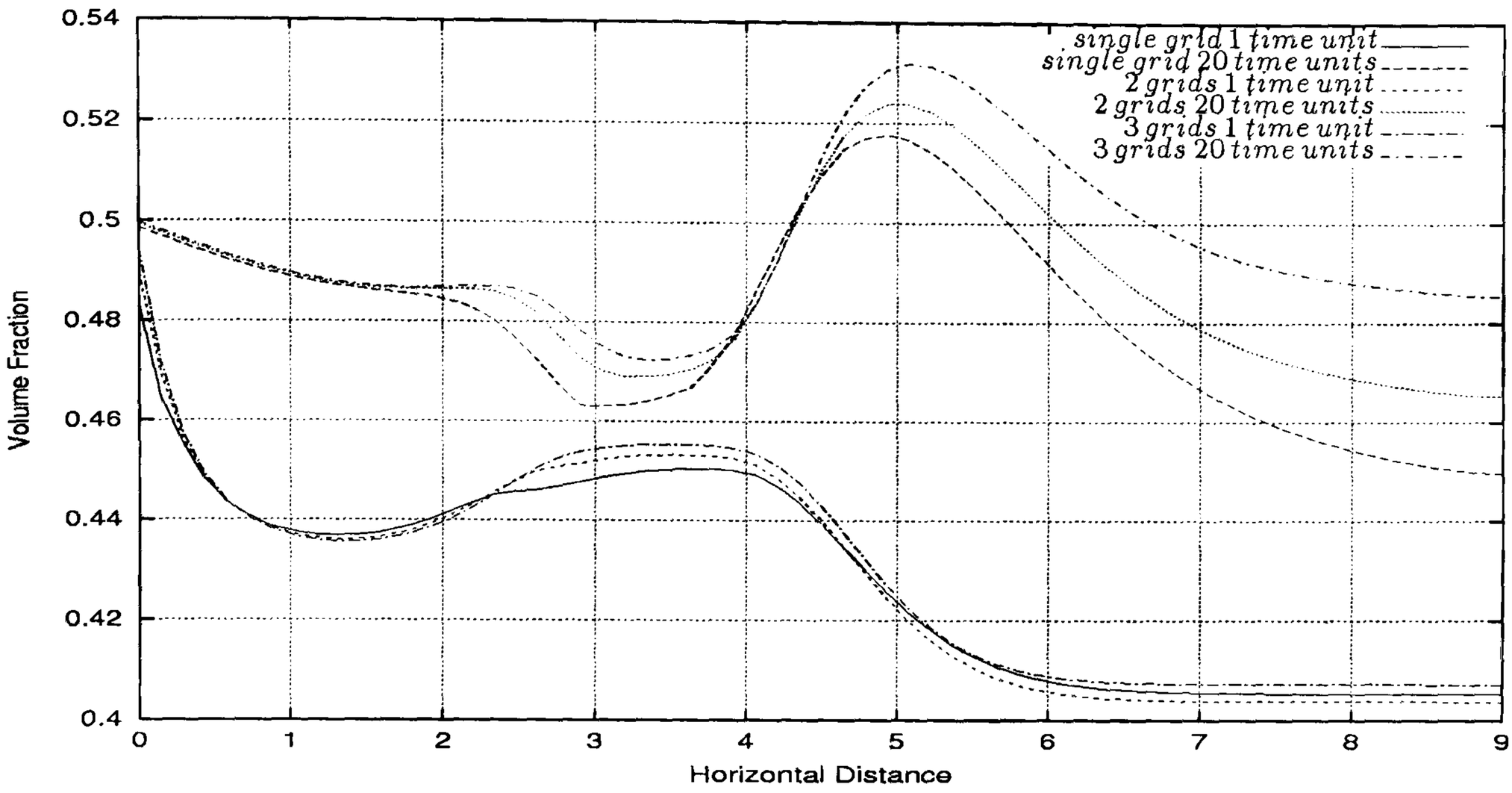


Figure 5.163: Multi-phase Contraction flow - volume fraction profile for *Phase II* along the line $y = 1.5$ for single 2 and 3 level computation. It can be observed good agreement between all the level computations. Note the small scale of the volume fraction axis.

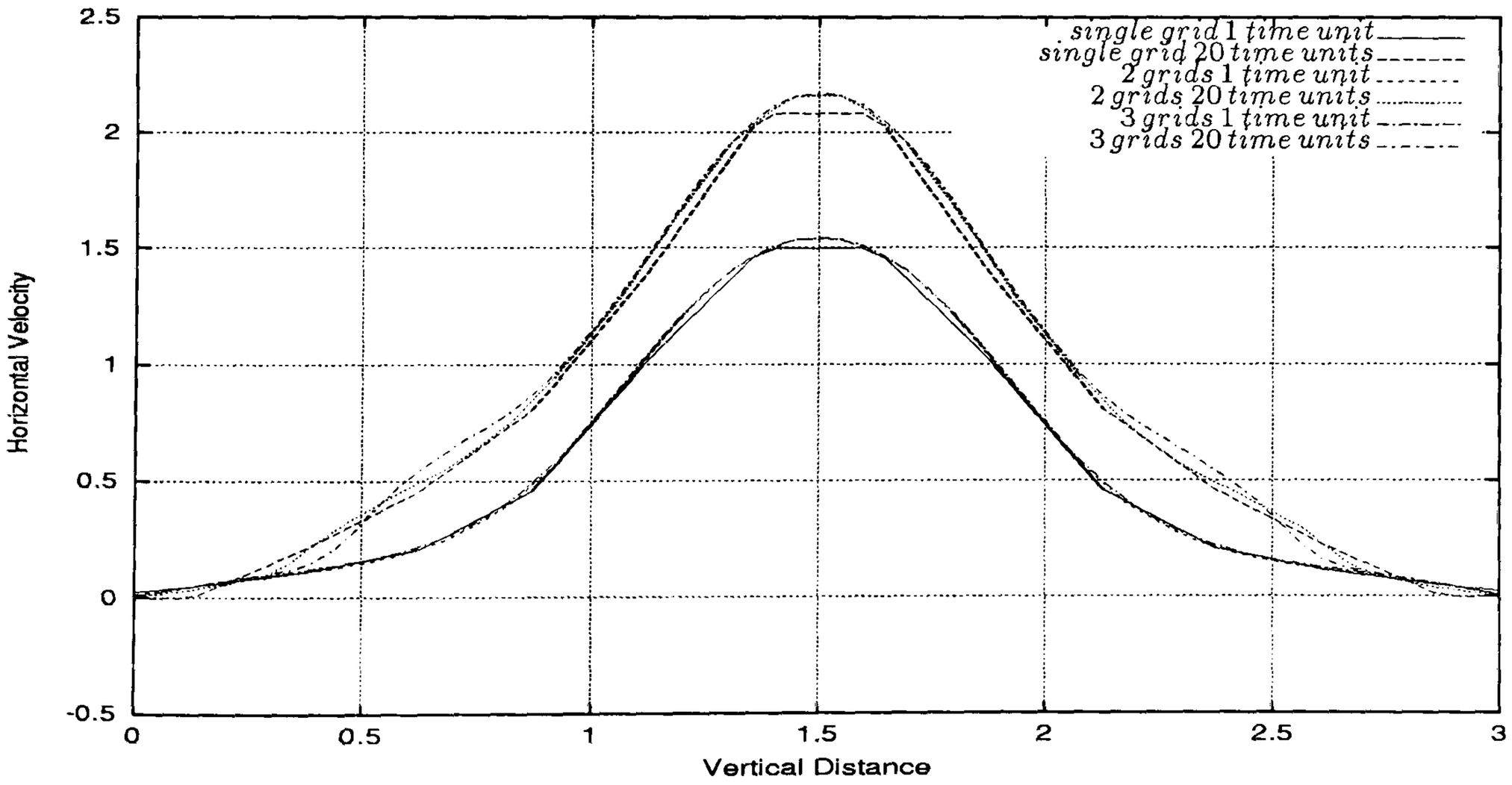


Figure 5.164: Multi-phase Contraction flow - horizontal velocity profile for *Phase I* along the line $x = 4.5$ for single 2 and 3 level computation. It can be observed good agreement between the parabolic profiles for all level computations.

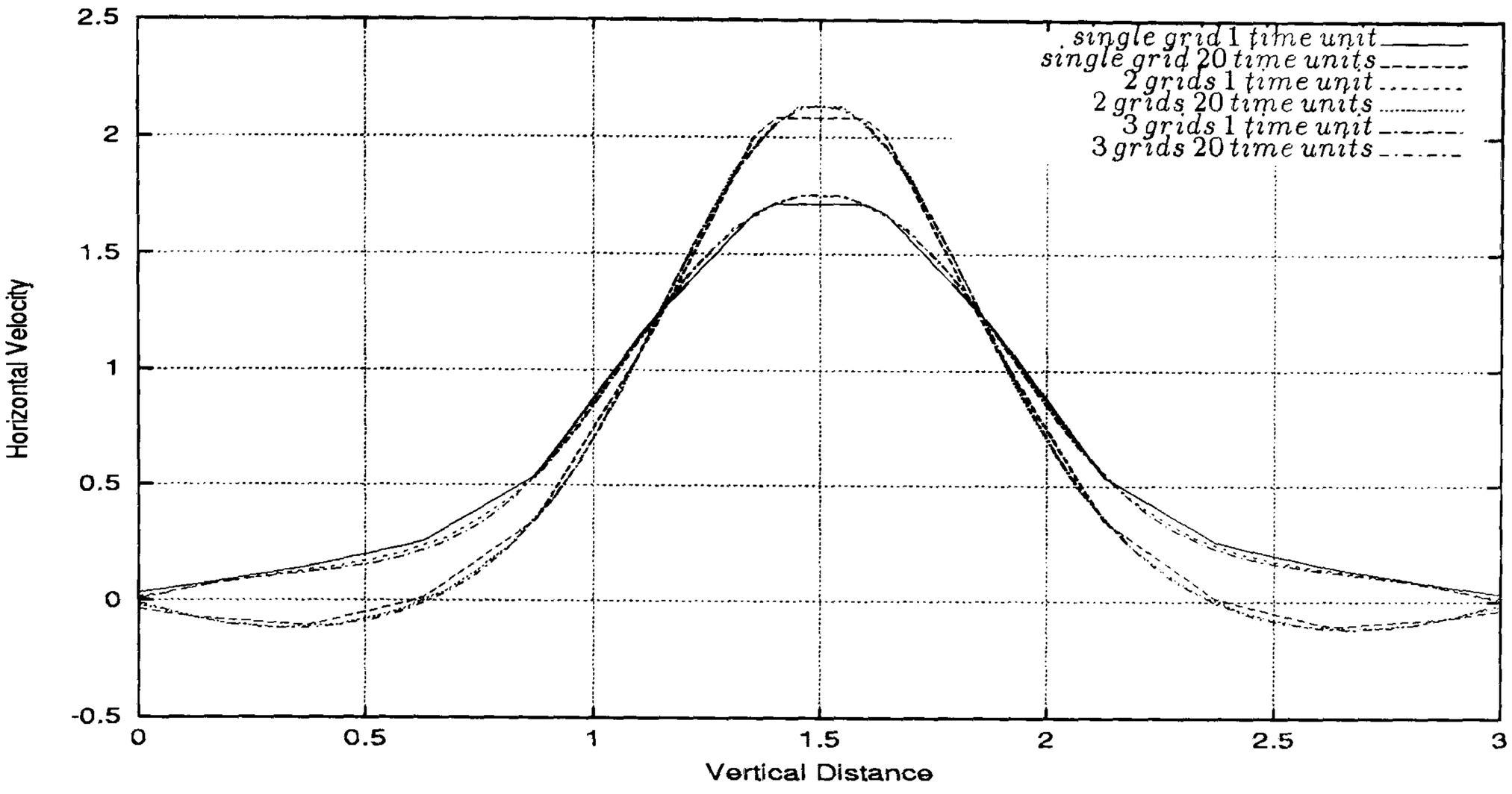


Figure 5.165: Multi-phase Contraction flow - horizontal velocity profile for *Phase II* along the line $x = 4.5$ for single 2 and 3 level computation. It can be seen good agreement between the parabolic profiles for all level computations.

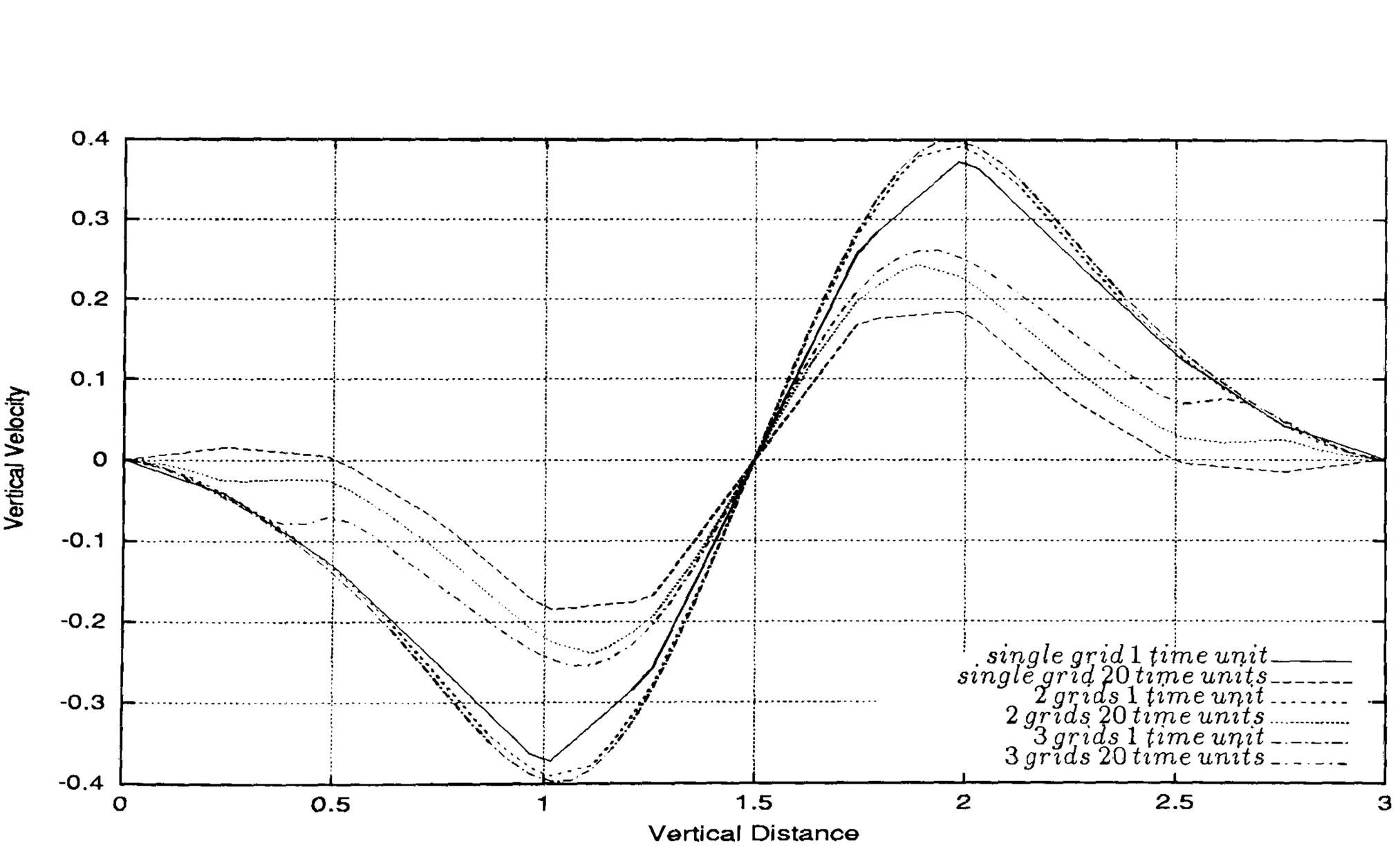


Figure 5.166: Multi-phase Contraction flow - vertical velocity profile for *Phase I* along the line $x = 4.5$ for single 2 and 3 level computation. It can be observed good general agreement between the profiles produced for all level computations.

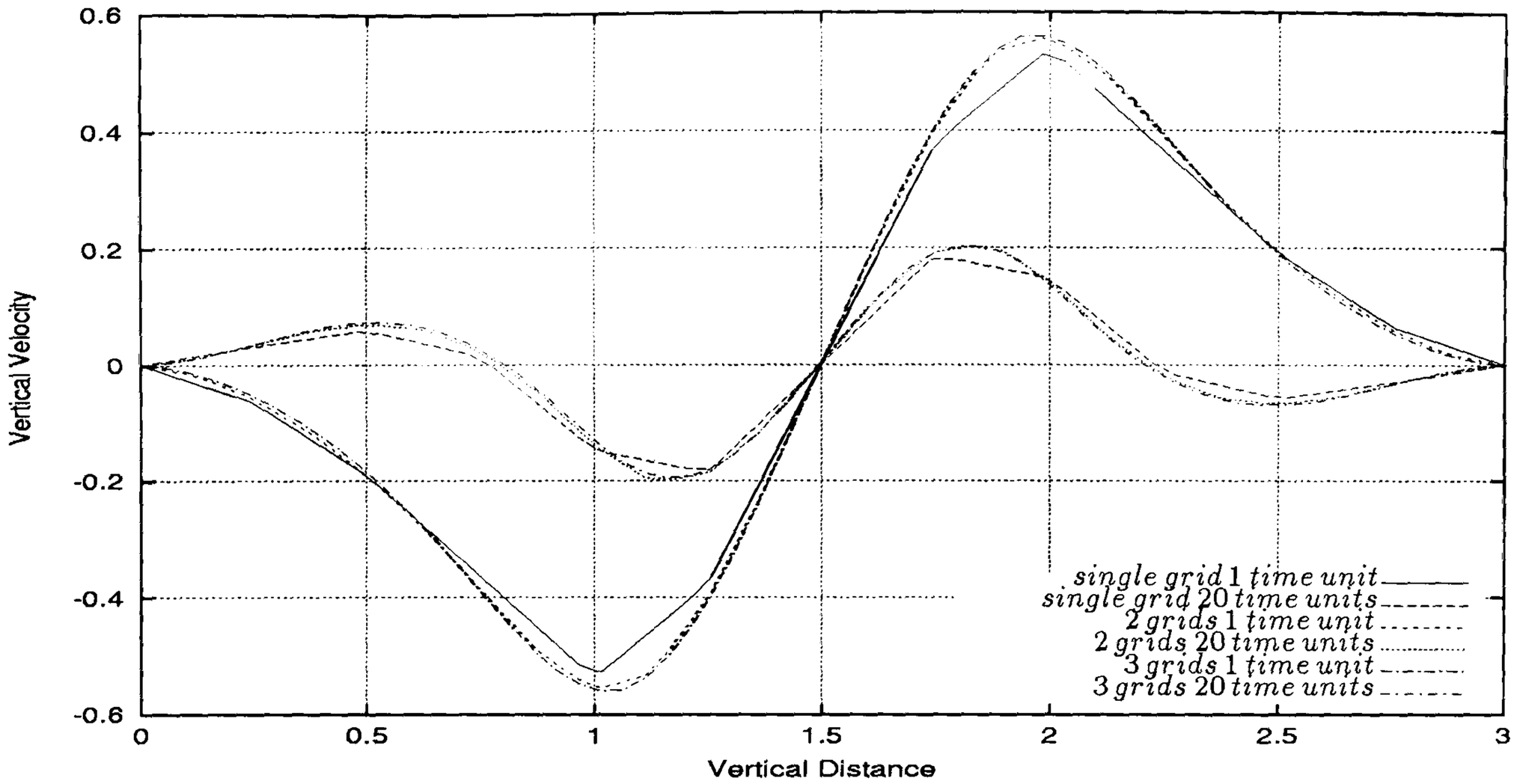


Figure 5.167: Multi-phase Contraction flow - vertical velocity profile for *Phase II* along the line $x = 4.5$ for single 2 and 3 level computation. It can be seen good agreement between the profiles produced for all level computations.

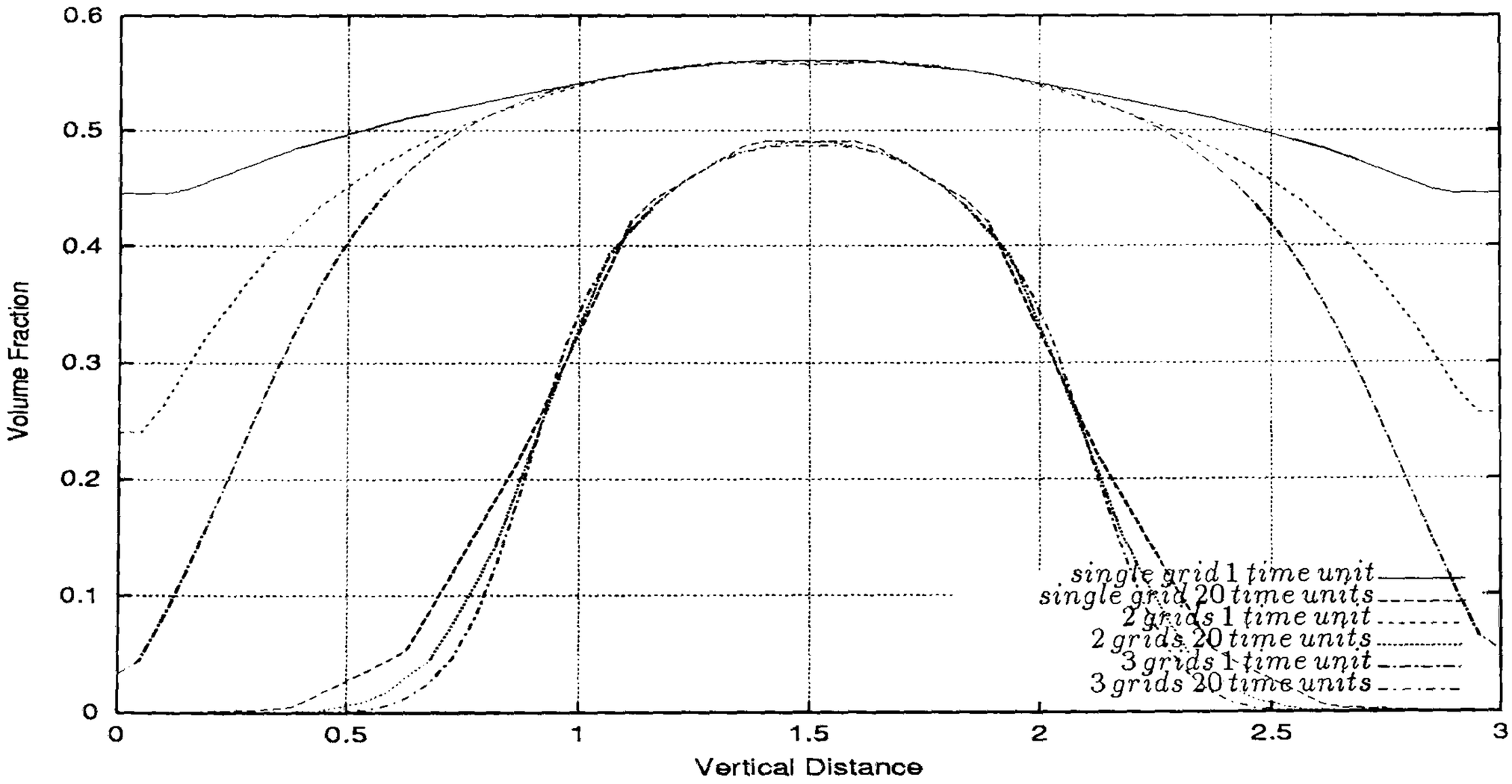


Figure 5.168: Multi-phase Contraction flow - volume fraction profile for *Phase I* along the line $x = 4.5$ for single 2 and 3 level computation. It can be observed good agreement between all level computations at 20 time units.

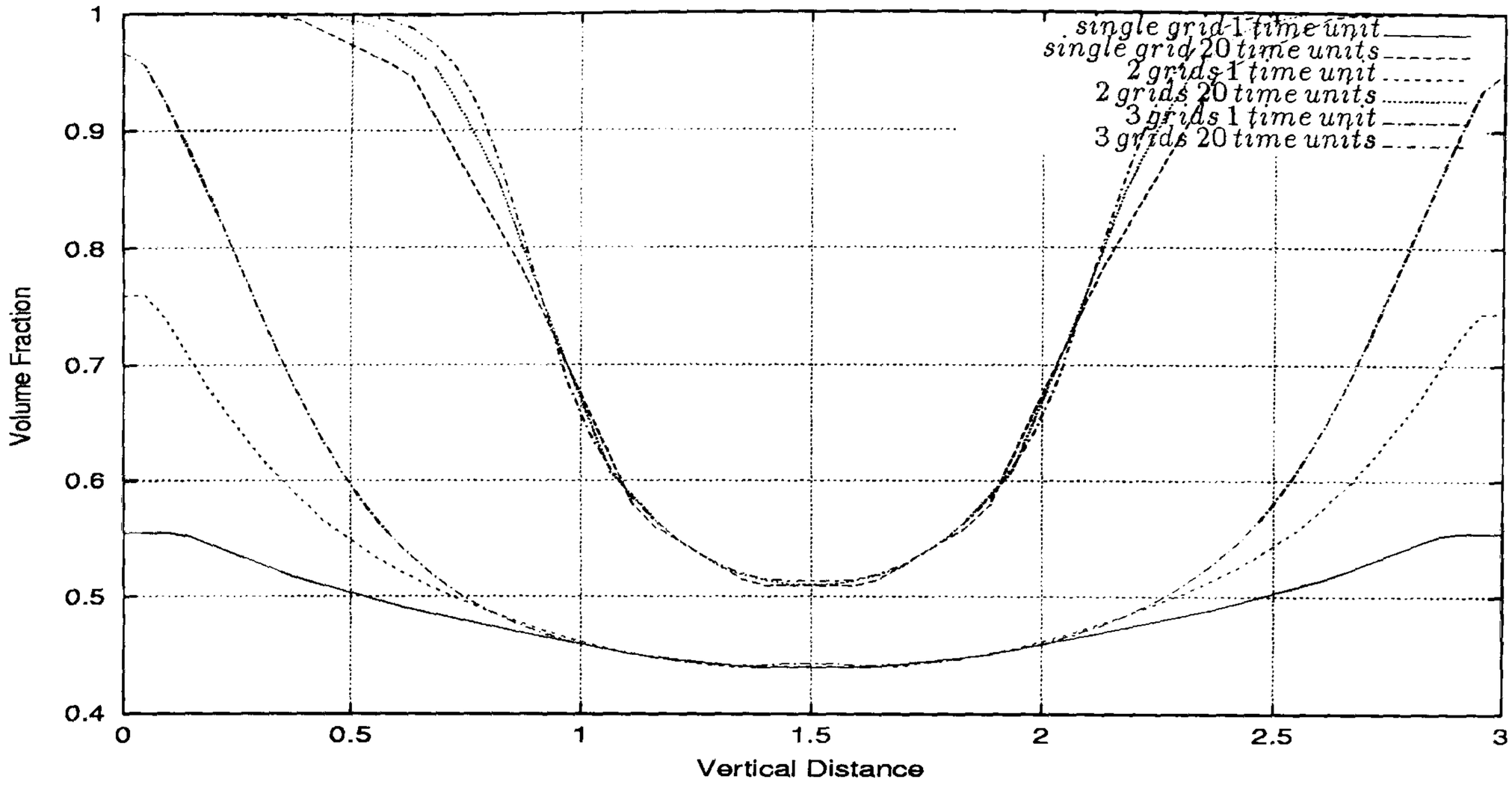


Figure 5.169: Multi-phase Contraction flow - volume fraction profile for *Phase II* along the line $x = 4.5$ for single 2 and 3 level computation. It can be observed good agreement between all level computations at 20 time units.

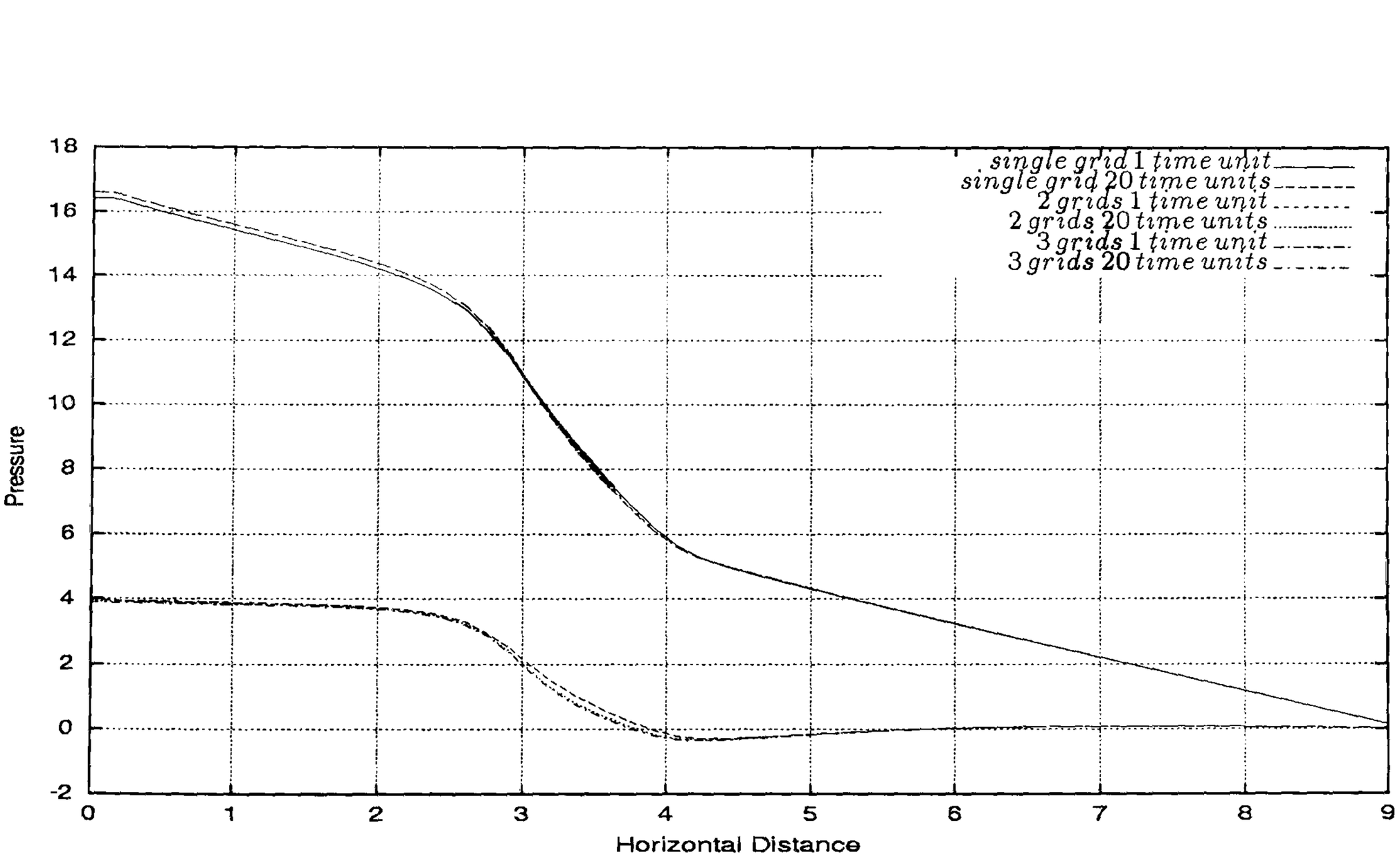


Figure 5.170: Multi-phase Contraction flow - pressure drop profile along the line $y = 1.5$ for single 2 and 3 level computation. It can be observed good agreement between the pressure drop values for all level computations.

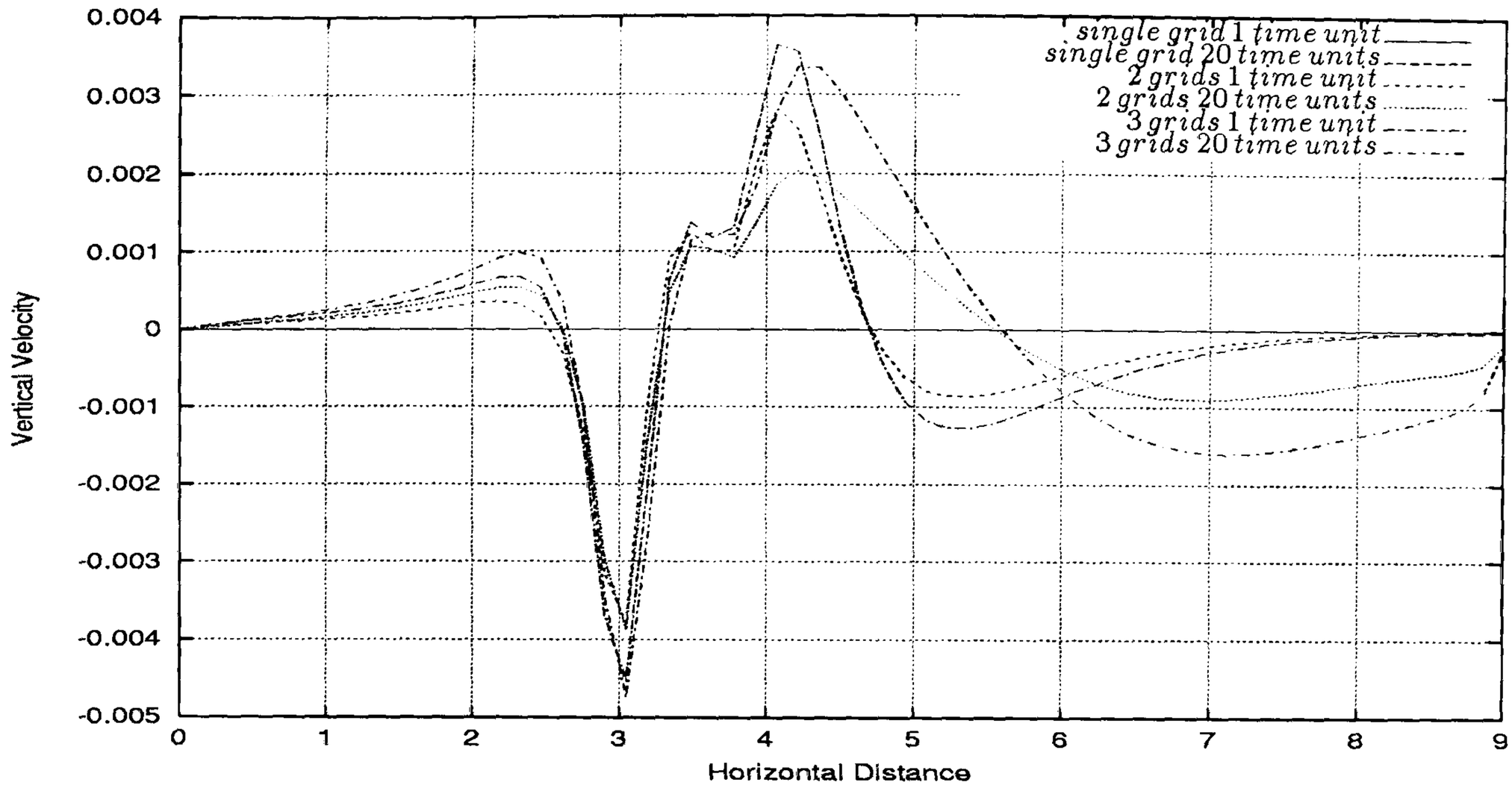


Figure 5.171: Multi-phase Contraction flow - vertical velocity profile for *Phase I* along the line $y = 1.5$ for single 2 and 3 level computation. It can be observed the same behaviour for the vertical velocity for 2 and 3 level computations. Note the horizontal line representing the computation in a single grid at 1 and 20 time units.

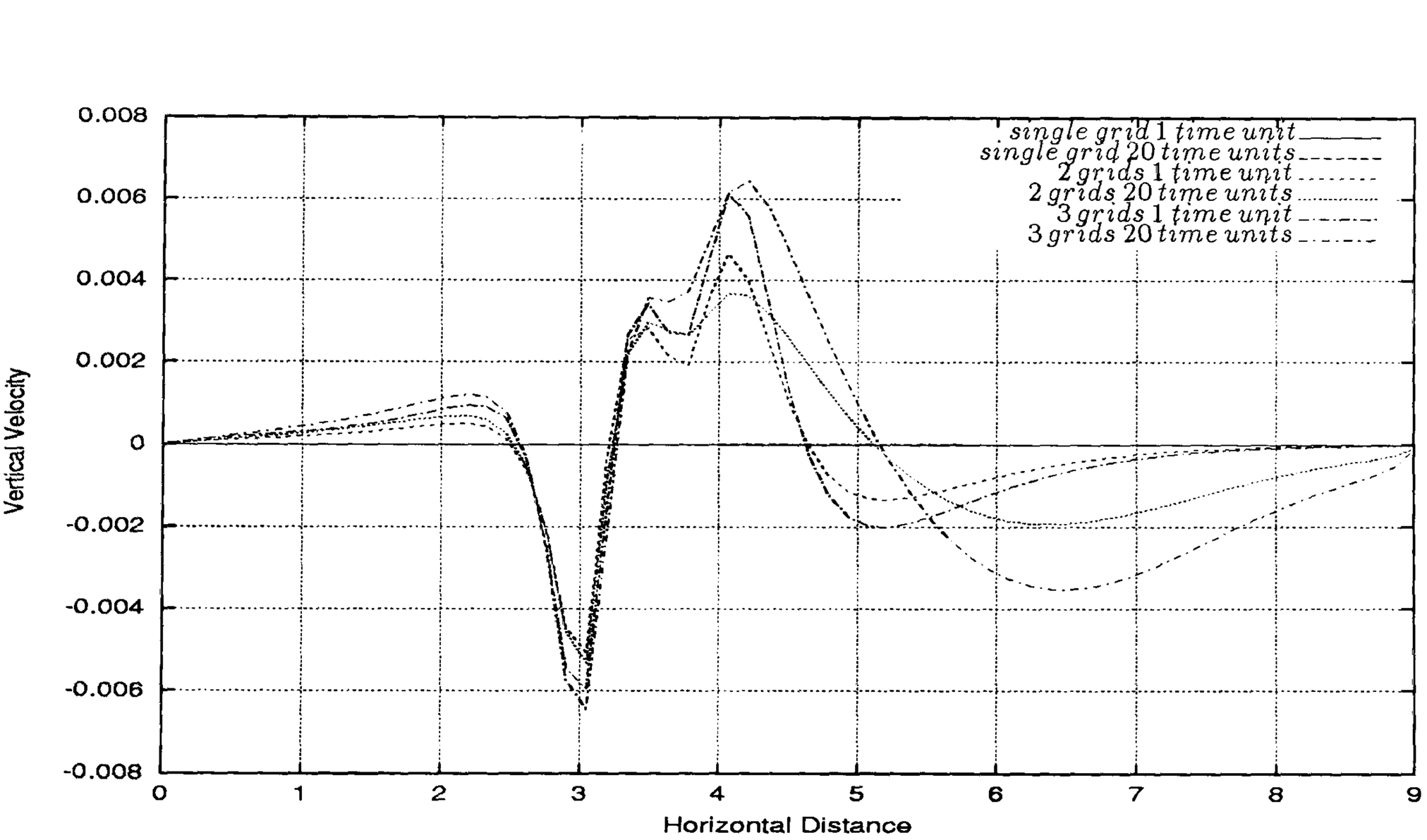


Figure 5.172: Multi-phase Contraction flow - vertical velocity profile for *Phase II* along the line $y = 1.5$ for single 2 and 3 level computation. It can be observed the same behaviour for the vertical velocity for 2 and 3 level computations. Note the horizontal line representing the computation in a single grid at 1 and 20 time units.

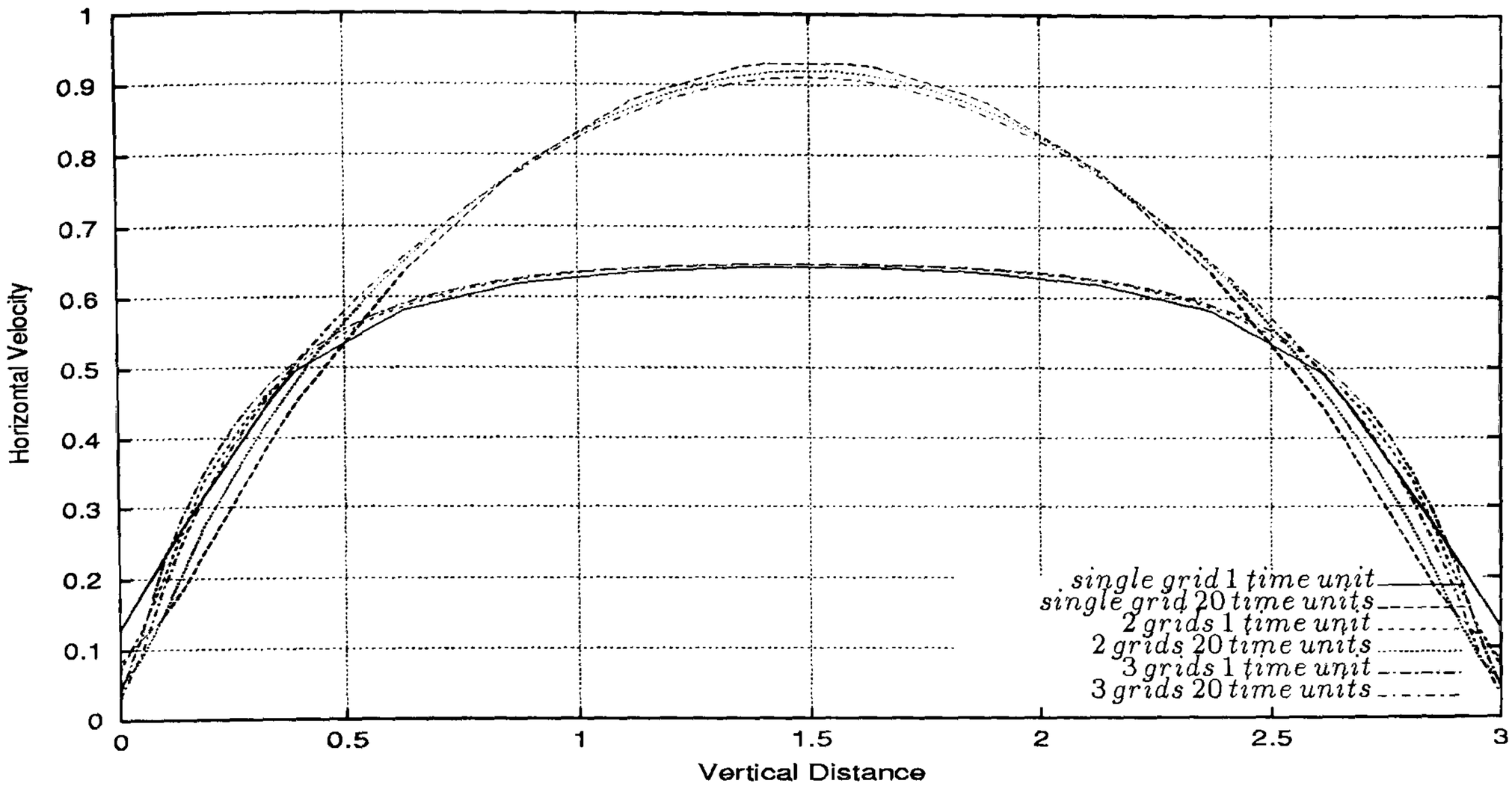


Figure 5.173: Multi-phase Contraction flow - horizontal velocity profile for *Phase I* along the line $x = 9.0$ for single 2 and 3 level computation. It can be seen good agreement between the fully developed parabolic profile for all level computations at 20 time units.

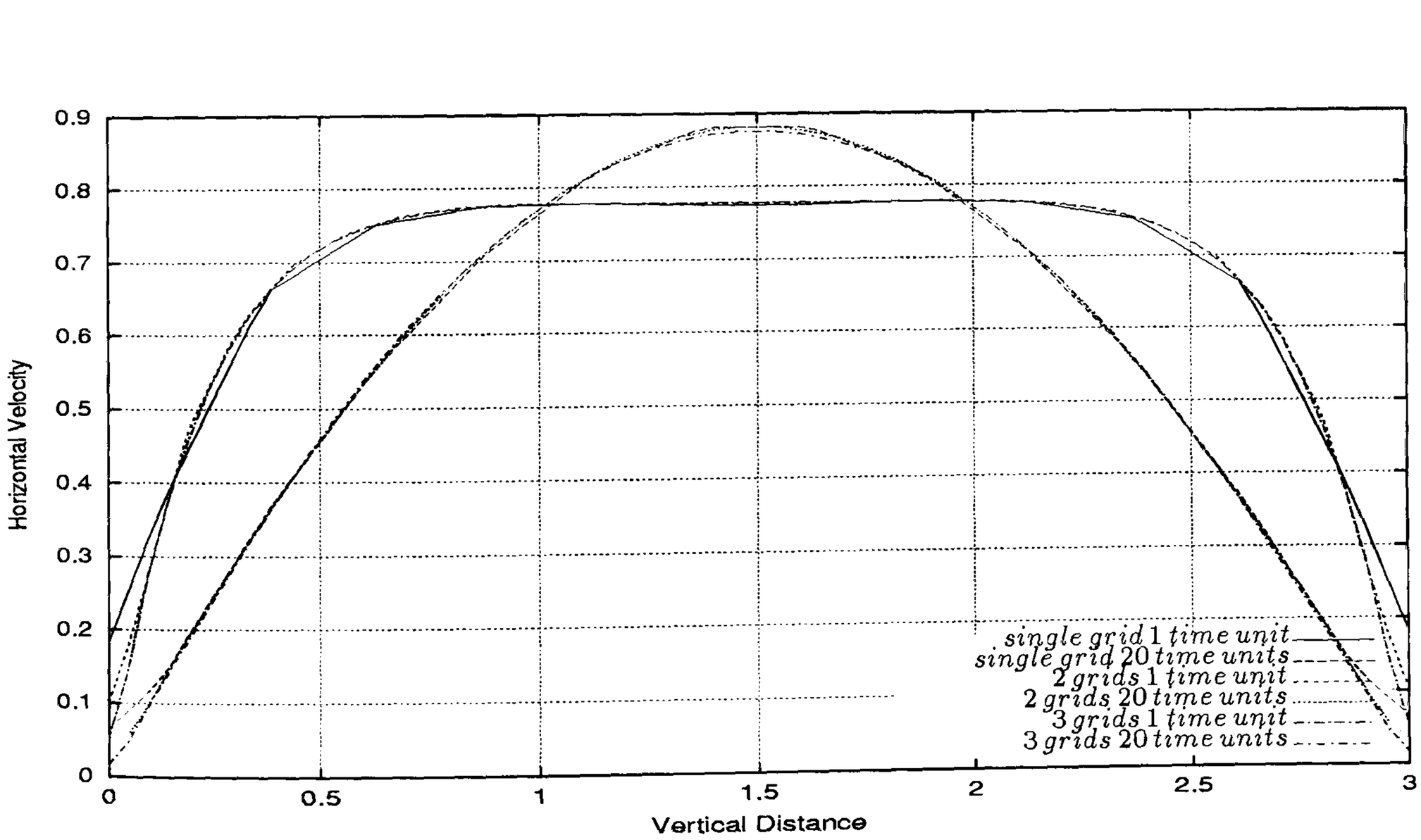


Figure 5.174: Multi-phase Contraction flow - horizontal velocity profile for *Phase II* along the line $x = 9.0$ for single 2 and 3 level computation. It can be seen good agreement between the fully developed parabolic profile for all level computations at 20 time units

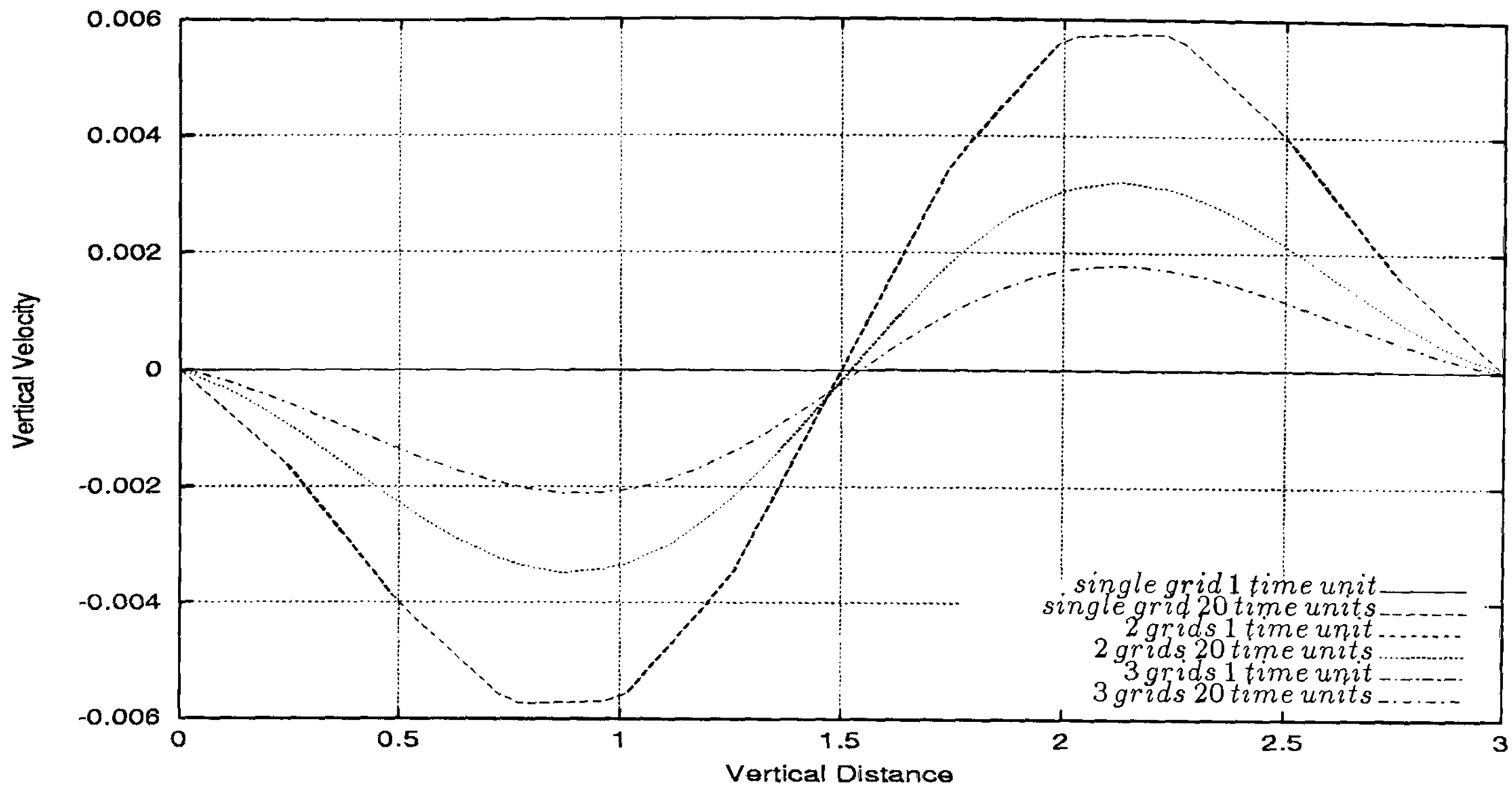


Figure 5.175: Multi-phase Contraction flow - vertical velocity profile for *Phase I* along the line $x = 9.0$ for single 2 and 3 level computation. It can be seen that the vertical velocities are very small and close to zero in all level computation.

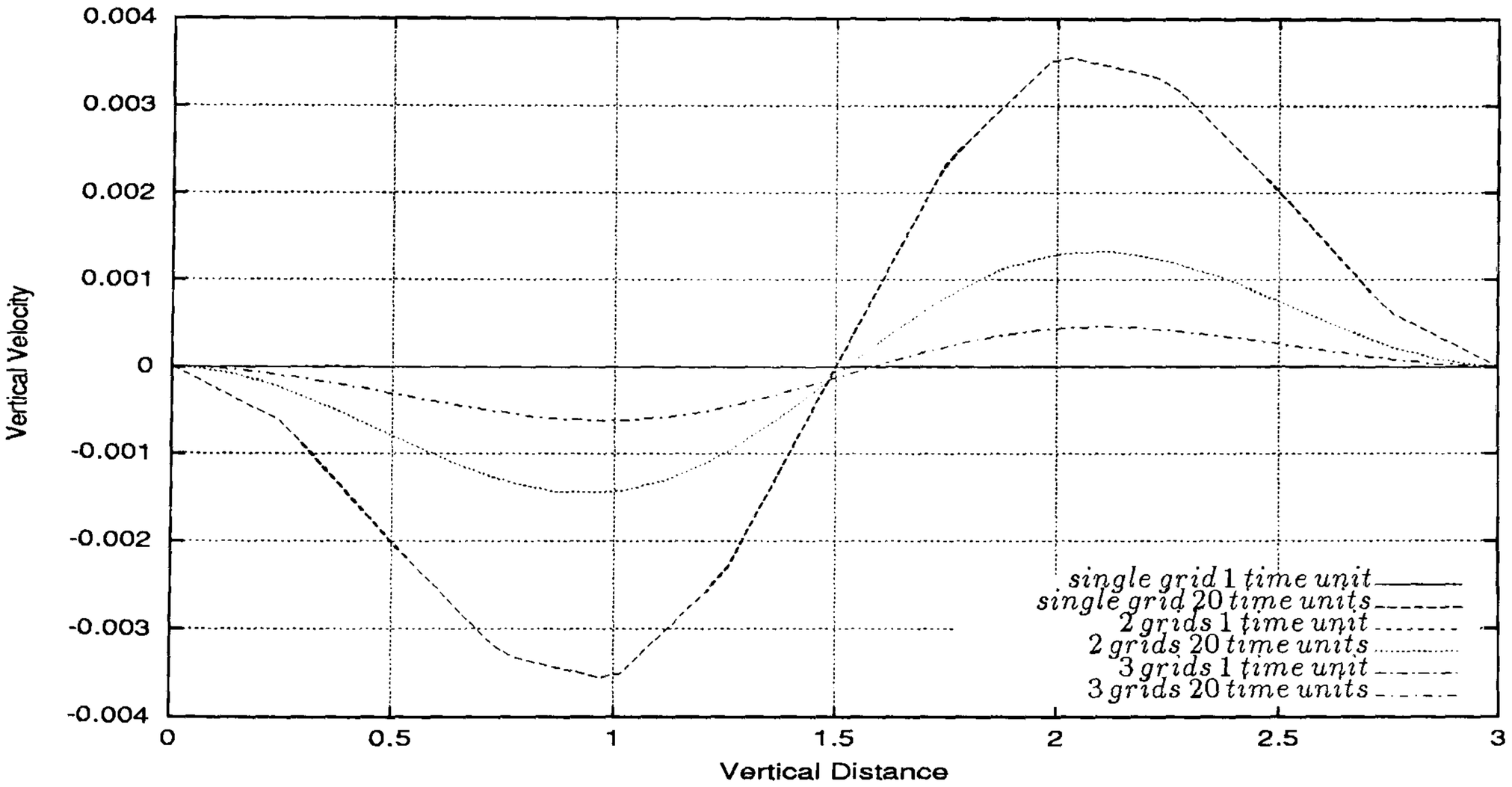


Figure 5.176: Multi-phase Contraction flow - vertical velocity profile for *Phase II* along the line $x = 9.0$ for single 2 and 3 level computation. It can be seen that the vertical velocities are very small and close to zero in all level computation.

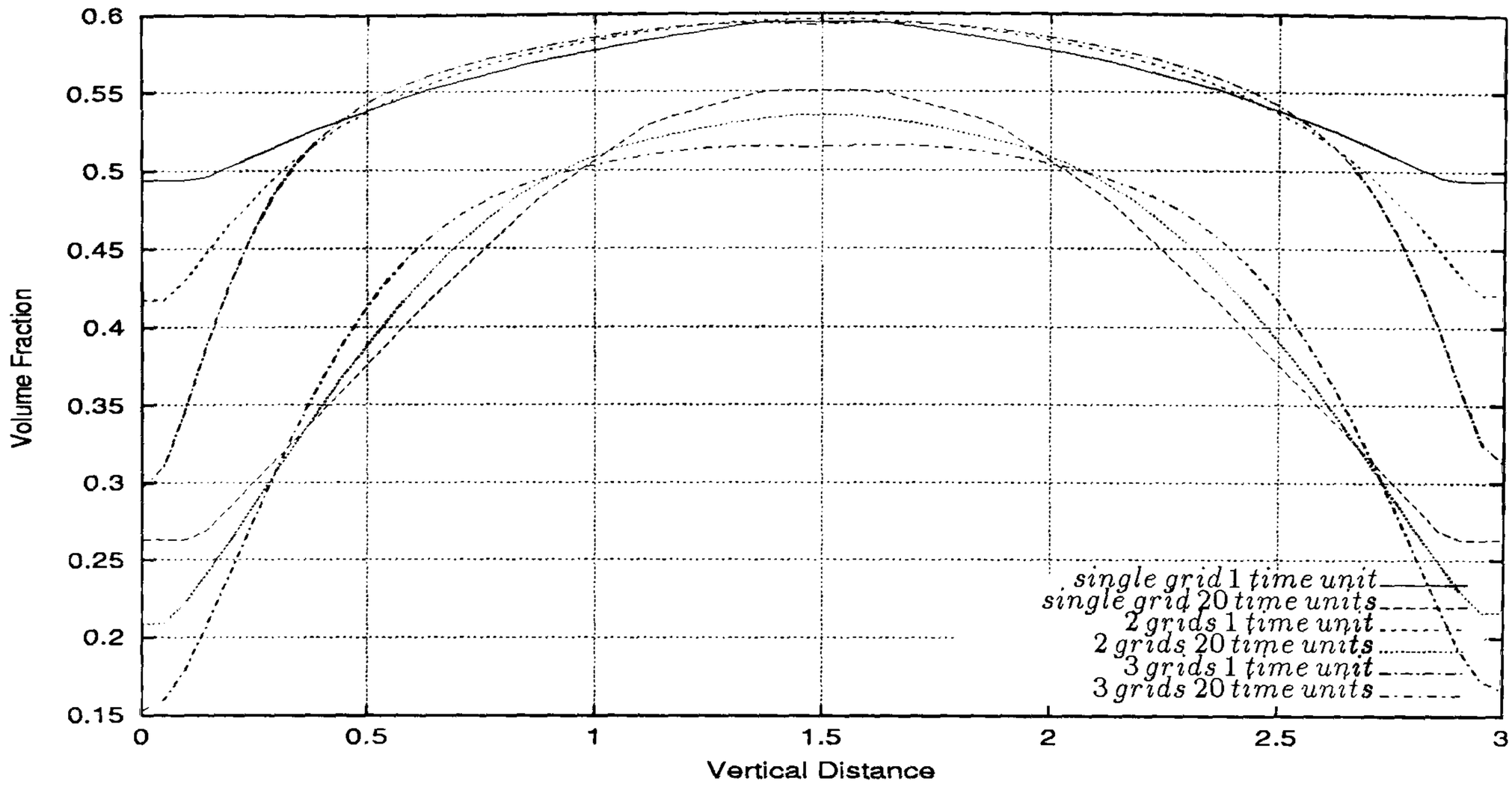


Figure 5.177: Multi-phase Contraction flow - volume fraction profile for *Phase I* along the line $x = 9.0$ for single 2 and 3 level computation. It can be seen that the disagreement in the walls reduces at 20 time units computational levels.

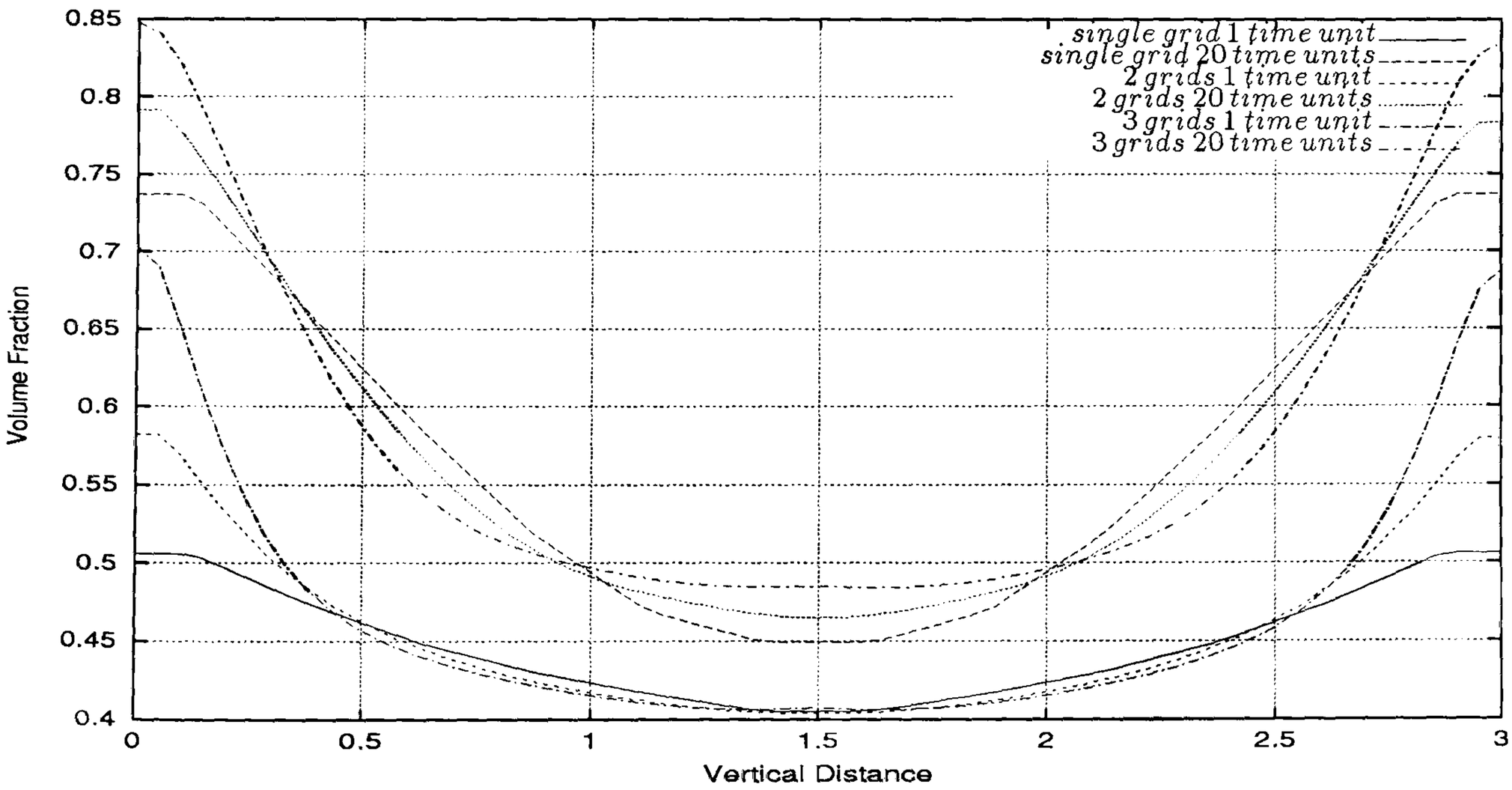


Figure 5.178: Multi-phase Contraction flow - volume fraction profile for *Phase II* along the line $x = 9.0$ for single 2 and 3 level computation. It can be seen that the disagreement in the walls reduces at 20 time units computational levels.

Comparison with CFX 4.3 Solutions

Figures 5.179 through 5.187 show the profiles of the horizontal velocity, vertical velocity, volume fraction and pressure drop along the lines $y = 1.5$, $x = 4.5$ and $x = 9.0$ for `transient Pamg Multi-phase` and `CFX 4.3` at 20 time units. In Figure 5.179 we present the horizontal velocity along the line $y = 1.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It is visible that in the curves corresponding to `CFX 4.3` and `transient Pamg Multi-phase` the horizontal velocity shows a good agreement for both phases.

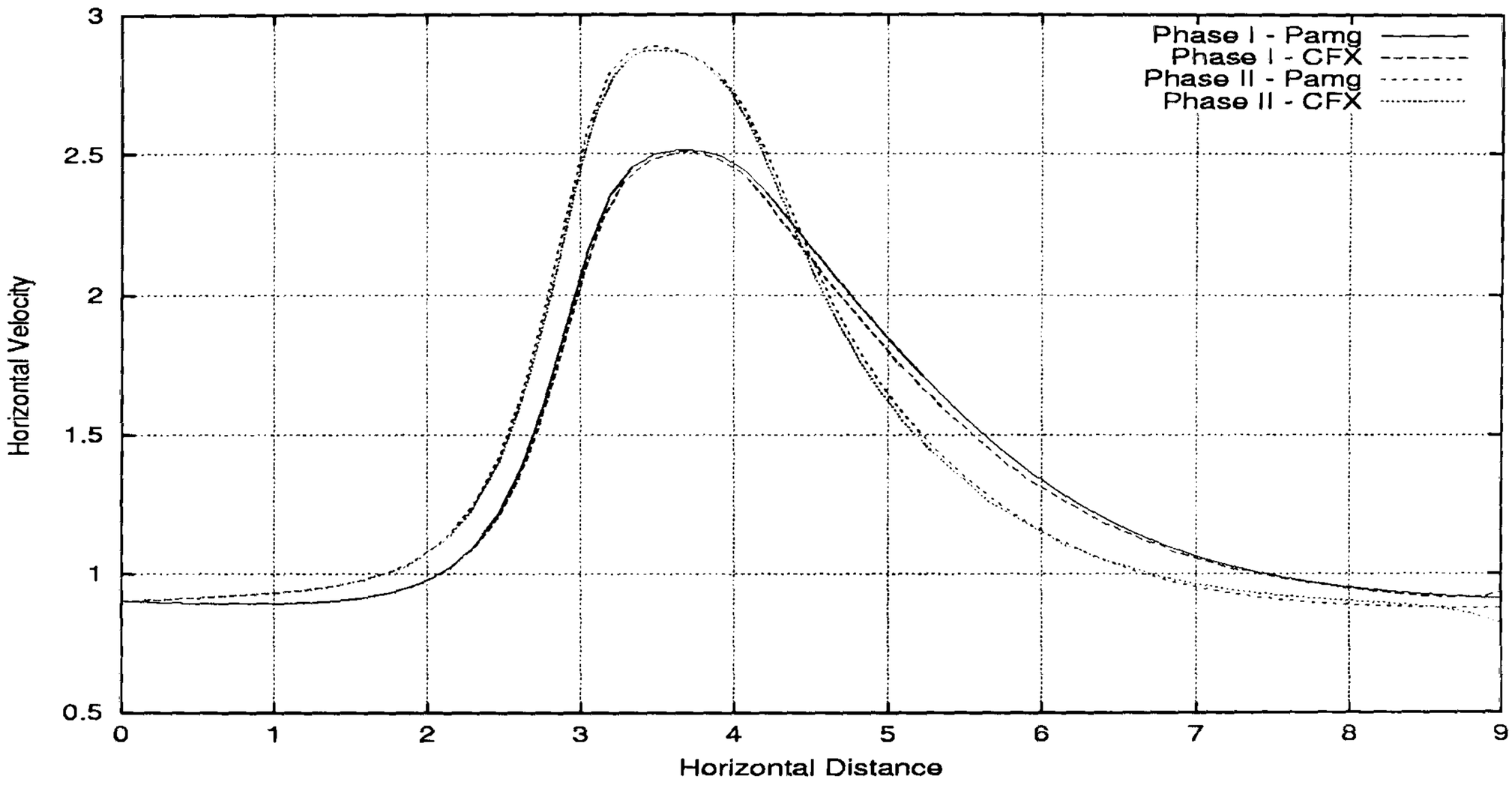


Figure 5.179: Multi-phase Contraction flow - comparison of horizontal velocity along the line $y = 1.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen good agreement between the computational results produced by both solvers.

Figure 5.180 presents the volume fractions along the line $y = 1.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. Good agreement is observed between the curves produced by both solvers in the first half of the horizontal distance. In the second half of the horizontal distance bigger differences between the curves are observed. However at the outlet these differences diminish.

In Figure 5.181 we present the vertical velocity along the line $y = 1.5$ for `transient Pamg Muti-phase` and `CFX 4.3` at 20 time units. It can be seen that the curves produced by `transient Pamg Multi-phase` exhibit a very similar pattern. However `CFX 4.3` produces curves that are very close to horizontal lines with vertical velocity null. The differences on this behaviour must be due to the different cell treatment of the velocities, which originate bigger differences in the contraction region. However the scale of the vertical velocity is very small.

Figure 5.182 shows the horizontal velocity along the line $x = 4.5$ for `transient Pamg Multi-phase` and `CFX 4.3` at 20 time units. The parabolic profile can be seen both for the `CFX 4.3` and `transient Pamg Multi-phase` results.

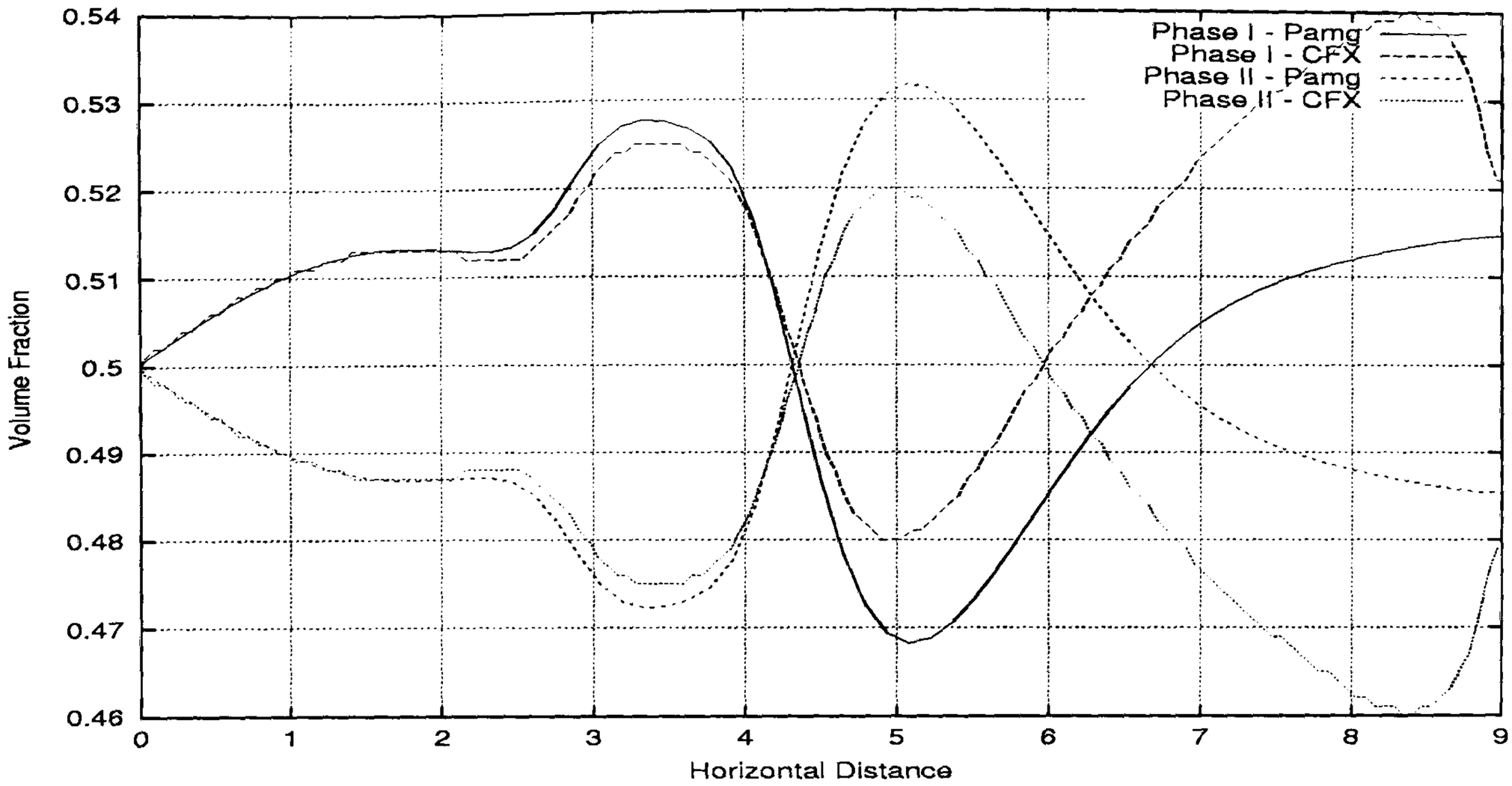


Figure 5.180: Multi-phase Contraction flow - comparison of volume fraction along the line $y = 1.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen good agreement between both solvers in the first half of the horizontal distance. In the second half some differences between the curves are observed. They diminish at the outlet.

In Figure 5.183 we present the vertical velocity along the line $x = 4.5$ for `transient Pamg Multi-phase` and `CFX 4.3` at 20 time units. It is visible the same pattern that the curves produced by both solvers exhibit.

Figure 5.184 present the volume fraction along the line $x = 4.5$ for `transient Pamg Multi-phase` and `CFX 4.3` at 20 time units. In general it can be observed a good agreement between the curves produced by both solvers. However at the inlet and outlet some substantial differences can be observed. This is due to different initial conditions for the volume fractions in `CFX 4.3`. In fact in `CFX 4.3` the initial volume fractions are settled to be 0.5 while in `transient Pamg Multi-phase` they are initialised by the volume fractions computed in the preceding time step.

Figure 5.185 presents the horizontal velocity along the line $x = 9.0$ for `transient Pamg Multi-phase` and `CFX 4.3` at 20 time units. The fully developed parabolic profile can be observed in the curves produced by both solvers.

In Figure 5.186 we present the vertical velocity profiles along the line $x = 9.0$ for `transient Pamg Multi-phase` and `CFX 4.3` at 20 time units. As we should expect in this outlet section the vertical velocity is very small. The curves produced by both solvers verify this and exhibit the same pattern over all the entire vertical distance.

Figure 5.187 presents the volume fraction along the line $x = 9.0$ for `transient Pamg Multi-phase` and `CFX 4.3` at 20 time units. The main differences that can be observed between the curves of both solvers are basically in the inlet and outlet sections. These, as we already explained above, are due to the different initial

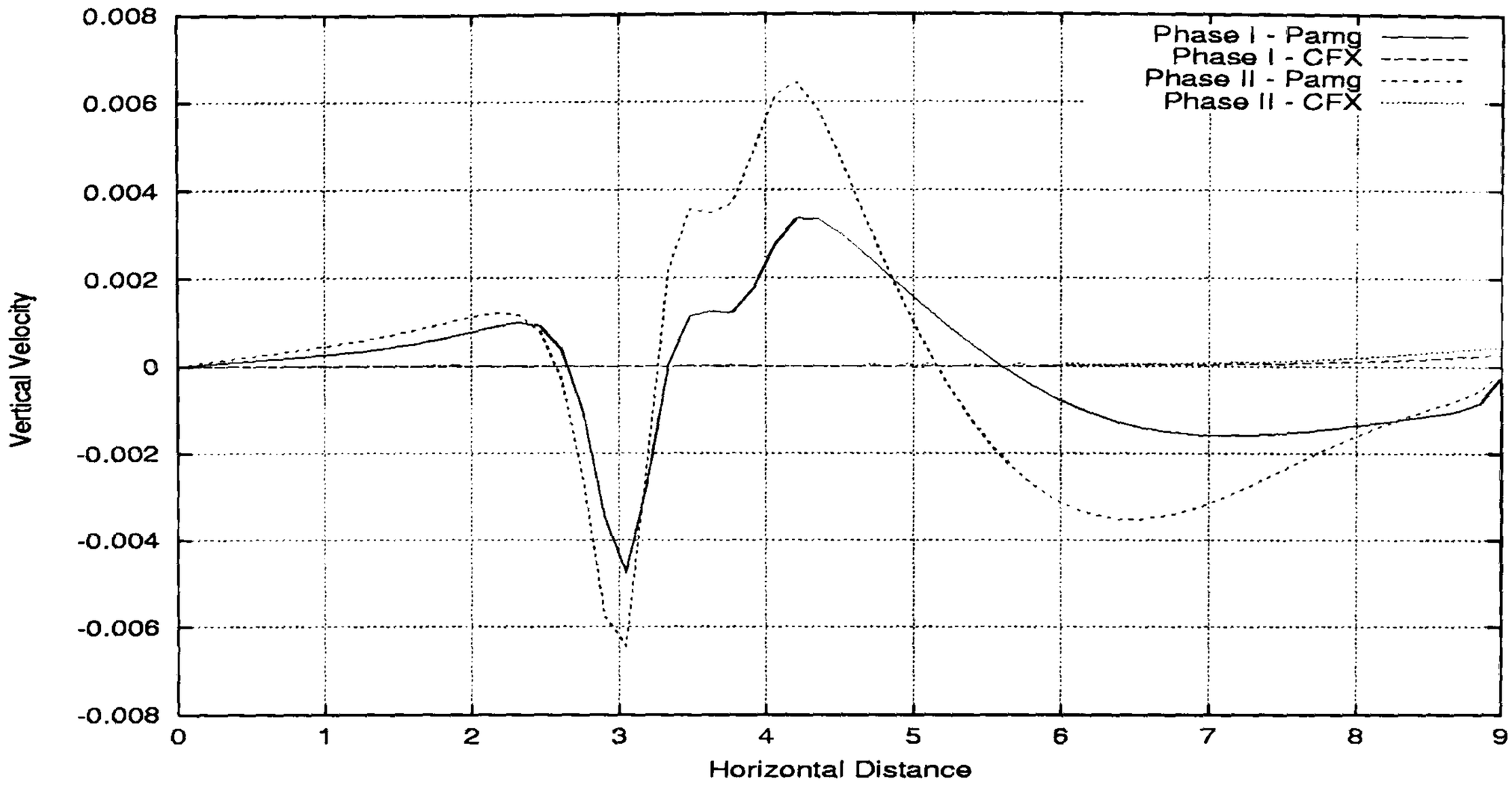


Figure 5.181: Multi-phase Contraction flow - comparison of vertical velocity along the line $y = 1.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen that the vertical velocities produced by `CFX 4.3` are very close to zero while `transient Pamg Multi-phase` produces positive and negative velocities around the contraction region ending with values close to zero in the outlet.

conditions used in the volume fractions.

Finally in Figure 5.188 we show the pressure drop along the line $y = 1.5$ for `CFX 4.3` and `transient Pamg Multi-phase`. The agreement that can be observed between the two curves is good. In fact only small differences are observed in the inlet and outlet regions. Since `CFX 4.3` and `transient Pamg Multi-phase` compute the unknowns in different places of each cell some slight differences are expected to be observed.

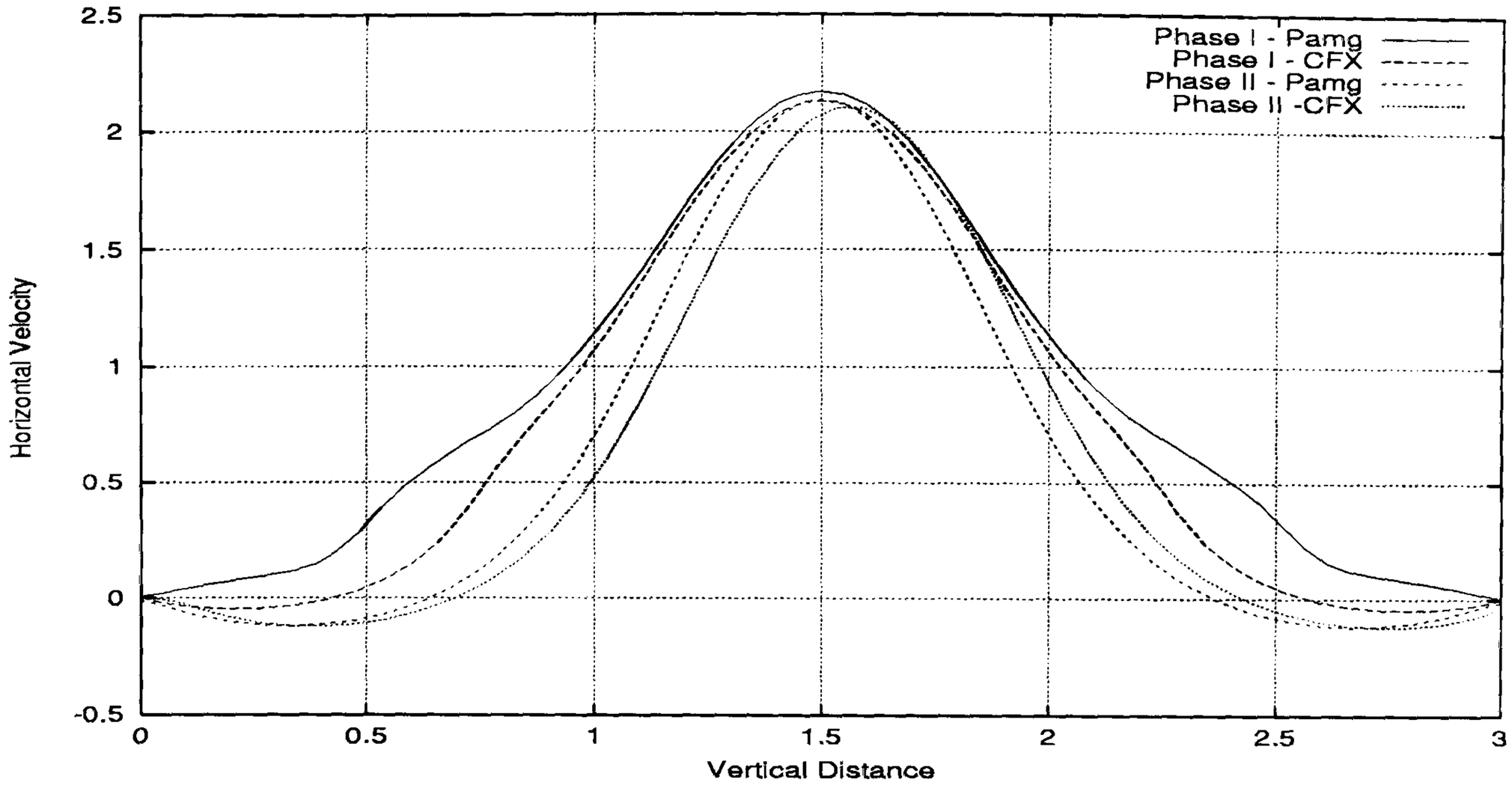


Figure 5.182: Multi-phase Contraction flow - comparison of horizontal velocity along the line $x = 4.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen good general agreement between the results produced by both solvers.

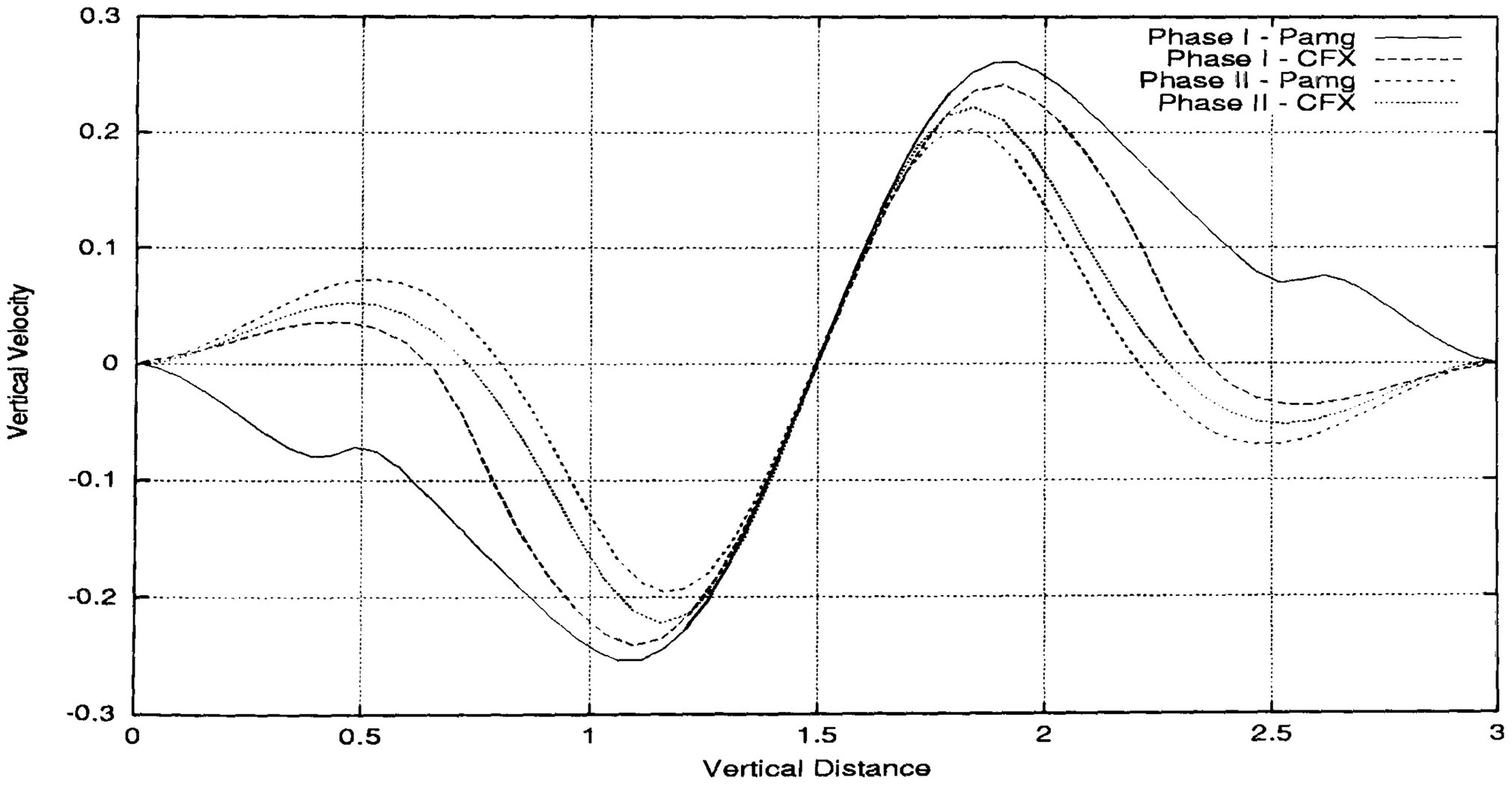


Figure 5.183: Multi-phase Contraction flow - comparison of vertical velocity along the line $x = 4.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be observed good general agreement between the vertical velocities produced by both solvers.

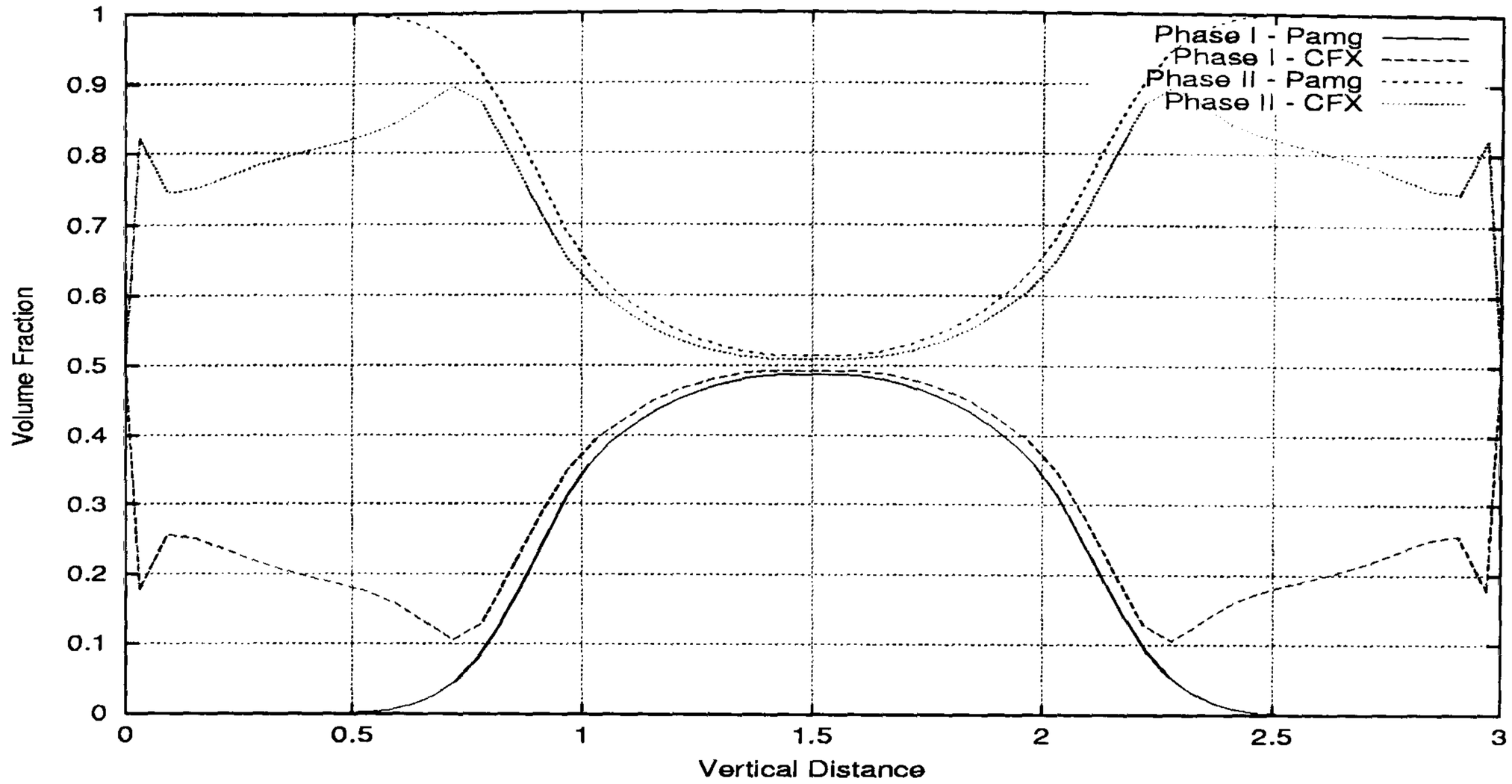


Figure 5.184: Multi-phase Contraction flow - comparison of volume fraction along the line $x = 4.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be observed the different initial volume fractions in the walls used by both solvers.

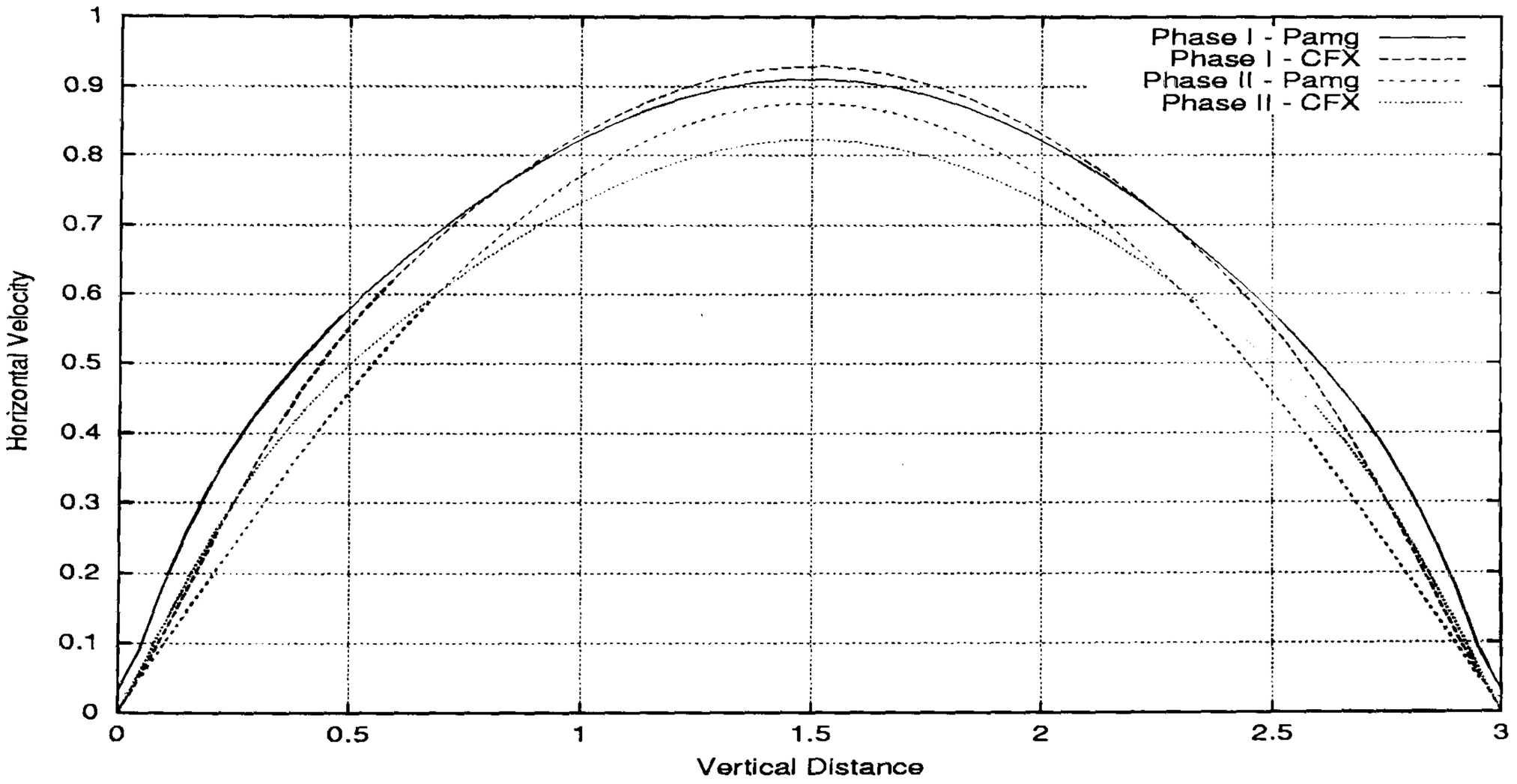


Figure 5.185: Multi-phase Contraction flow - comparison of horizontal velocity along the line $x = 9.0$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be observed good general agreement between the parabolic profiles developed by both solvers.

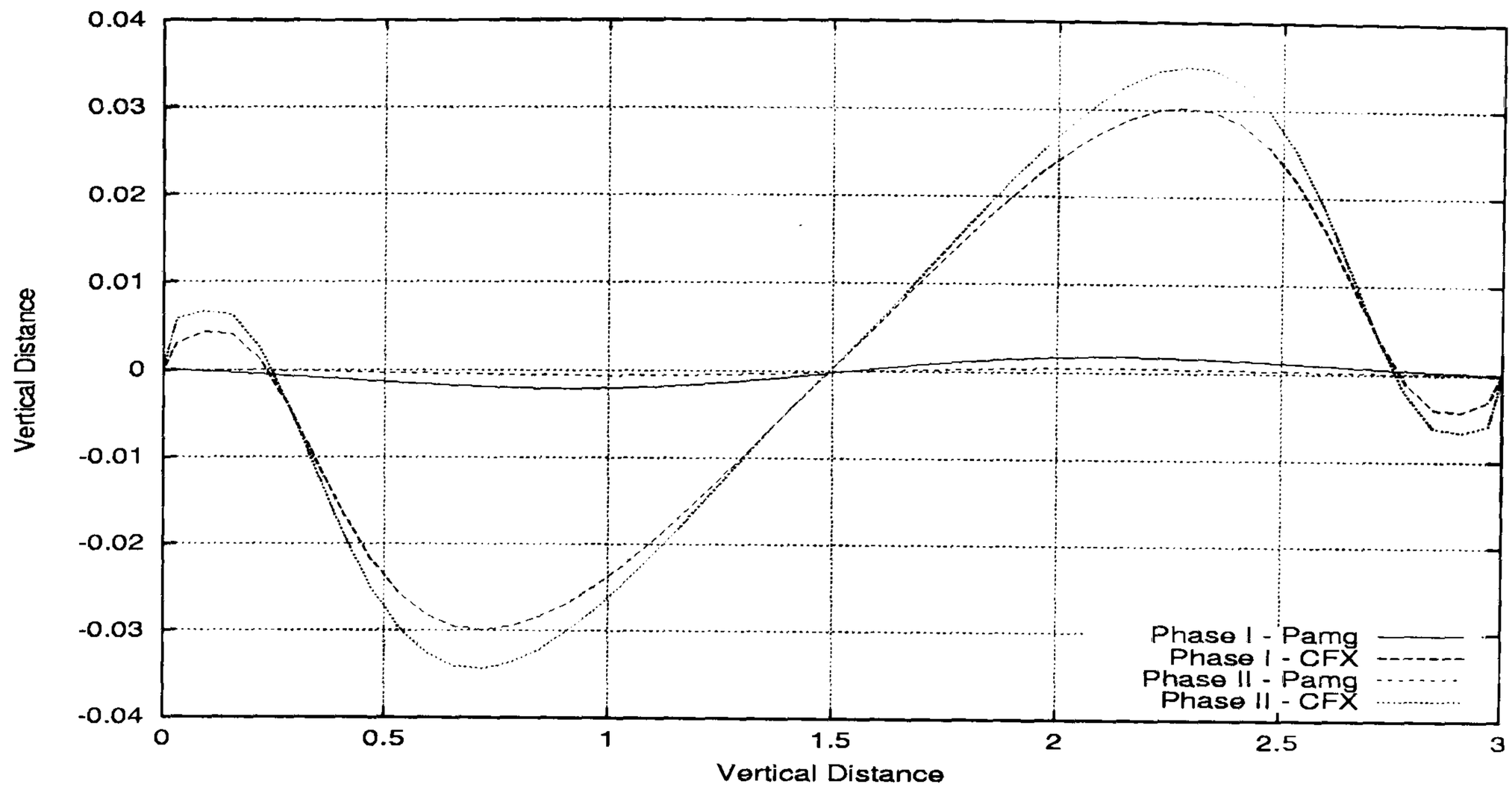


Figure 5.186: Multi-phase Contraction flow - comparison of vertical velocity along the line $x = 9.0$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen good agreement between the results produced by both solvers.

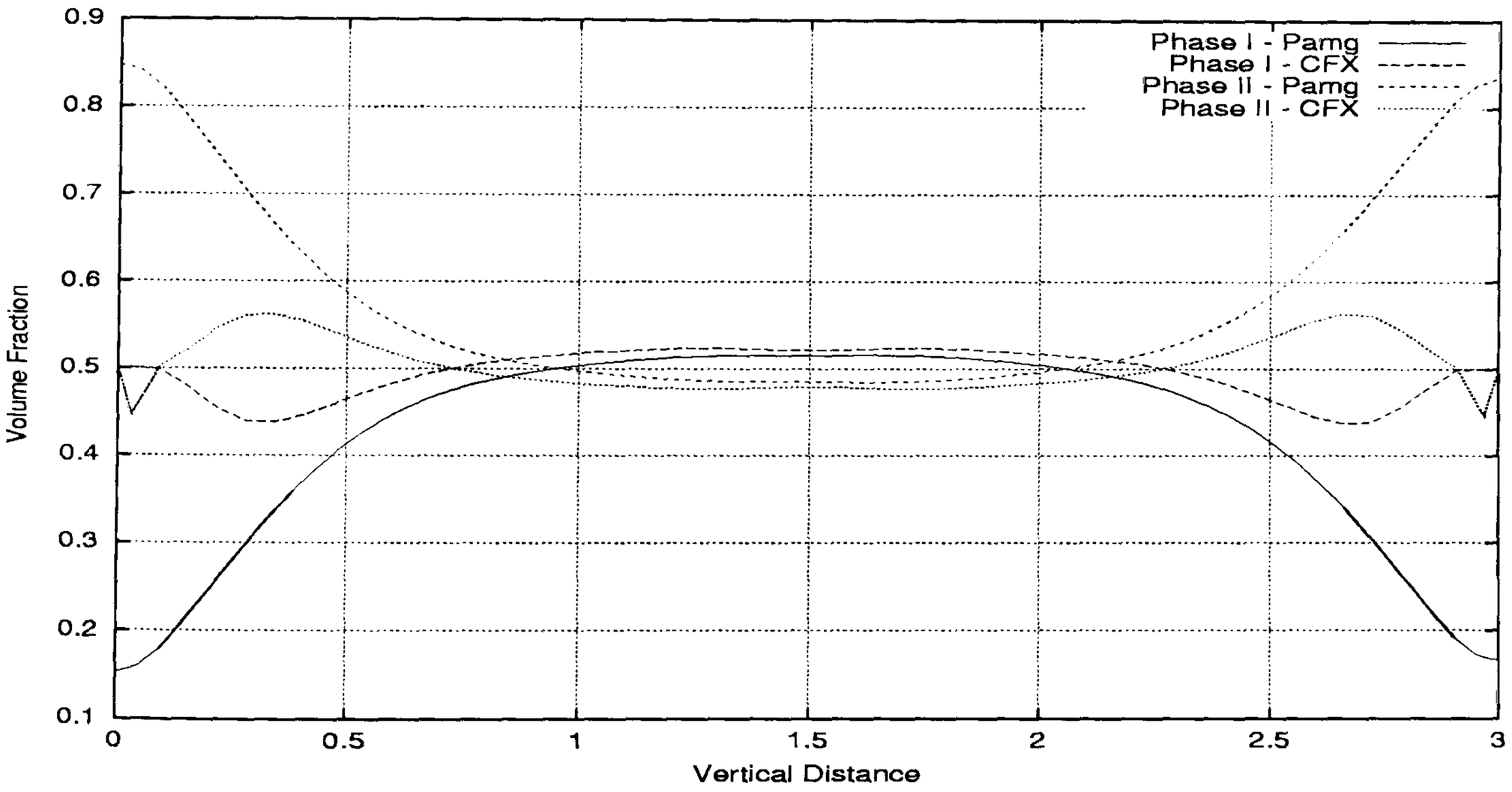


Figure 5.187: Multi-phase Contraction flow - comparison of volume fraction along the line $x = 9.0$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be observed the different initial volume fractions in the walls used by both solvers.

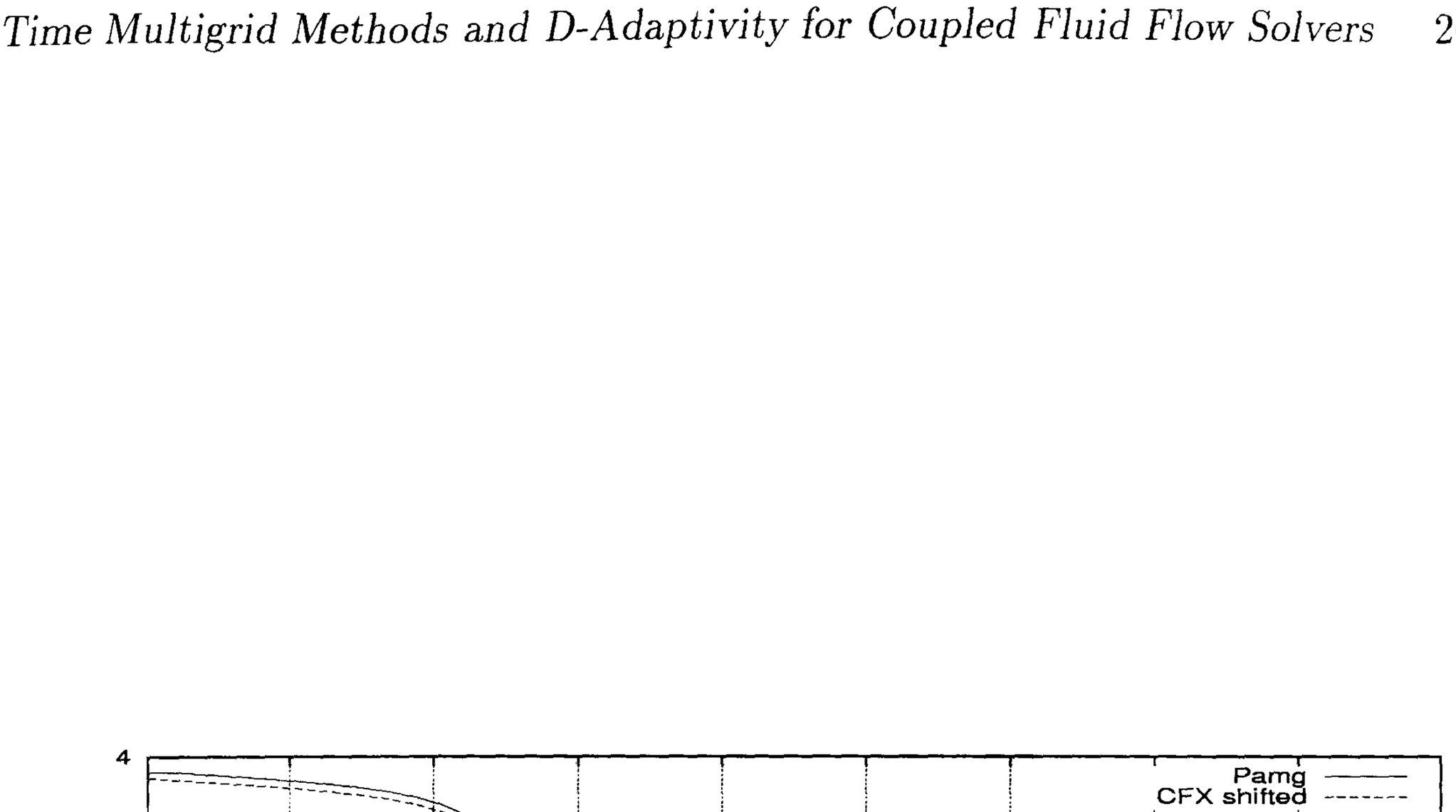


Figure 5.188: Multi-phase Contraction flow - comparison of pressure drop profiles along the line $y = 1.5$ for `CFX 4.3` and `transient Pamg Multi-phase` at 20 time units. It can be seen good agreement between the pressure drops measured by both solvers.

Summary

To end the validation process of the `transient Pamg Multi-phase` we simulate a two-phase fluid flow through a contraction. This problem is even more difficult than the two-phase fluid flow through a T-junction. The main reason is its geometry constraint. This originates big differences in the horizontal velocities and regions where the vertical velocities change from positive to negative values very quickly. Consequently the solution algorithm has been deeply tested. Another important point is the quality of the grid chosen. Evidence suggests that only with a very fine grid it is possible to reach the solution with a reasonable cost. Like in the T-junction problem two-different fluids with different viscosities and densities have been used in this simulation. Important sections have been chosen to present the solution profiles. Results for 3 level computations have been presented. The grid independence of the solutions have been verified by comparing results for single 2 and 3 level computations. The time step independence has been more difficult to verify. In the authors opinion the main reason for that is related to the bigger variations observed in the velocities. Choosing bigger time steps the variations on the velocities become even bigger and originate some singular jacobian matrices. Consequently the flow behaviour becomes more unsteady and consequently the time accuracy is more difficult to be validated.

To finish the validation process of this problem results produced by `CFX 4.3` have been compared with the results produced by the `transient Pamg Multi-phase` solver. General good agreement between the results produced by both solvers has been observed. However the different location of the unknowns in the control volumes used by `transient Pamg Multi-phase` and `CFX 4.3` and the initialisation conditions for the volume fractions used by both solvers originate some differences.

5.4 The performance of Transient Pamg.

In Sections 5.2 and 5.3 we established the accuracy of the solutions provided by `transient Pamg Single-phase` and `transient Pamg Multi-phase`. Particular focus has been undertaken on the study of grid independence and time step independence of the solutions for the several test cases presented. Furthermore, the solutions produced for both solvers have been compared with the solutions given by the commercial code `CFX 4.3`.

In this section we will study the performance of both solvers. This will be done in terms of numerical convergence and robustness. Numerical convergence is the property that the difference between the exact solution of the discrete equations and the computational solution is small. This computational error is usually measured by the reduction of the norm of the residual. Robustness is the property that the convergence is independent of the quality of the initial guess.

Starting with the robustness, `transient Pamg Single-phase` and `transient Pamg Multi-phase` verify this property since, as we saw in the sections above, results presented for the several test cases where obtained always with simple and

standard initial guesses. In fact, the velocity and pressure fields were set to zero and the volume fractions to 0.5 for each phase of the multi-phase test cases.

The numerical convergence is normally measured by the convergence factor. In order to reduce the influence of the initial guess we use the asymptotic value of the convergence factor.

We introduce the concept of work unit[10], which is the cost of one relaxation sweep on the finest grid on which the solution is sought. The convergence of `transient Pamg Single-Phase` and `transient Pamg Multi-phase` is measured in terms of an average residual which is equal to[10] the L_2 norm of the residual normalised by the number of discrete equations on the finest grid:

$$R = \sqrt{\frac{\sum_{ij} \sum_{\alpha=1}^{M} \sum_{k}^{3} (f_{ijk}^{\alpha})^2}{3NM}}$$

where i and j refer to a particular computational cell, k is the number of equations defined on a particular cell phase, α the phase, N is the number of computational cells and M is the number of phases. The value of R computed for each iteration in the finest grid is compared with the given tolerance 10^{-6} for each time step. If it is smaller than the required tolerance the simulation ends for that time step.

We remark that the multigrid method significantly accelerates the speed of convergence of the single-grid.

In Tables 5.1 through 5.6 we present the results of the work units (per time step) spent in the simulation of the test cases at 1, 5, 10 and 20 time units for different number of grids. It can be seen that the cost for reaching the solution above the required tolerance, diminuish 1/3 when changing from single grid to 2 grids computations and from 2 to 3 level computations.

Time	single grid	2 grids	3 grids	4 grids	5 grids
1	600	172	103	88	92
5	400	151	77	63	58
10	400	148	75	64	59
15	400	159	77	64	59
20	400	160	77	64	59

Table 5.1: Single-phase channel flow - work units per time step for different number of grids

The cycles used in all the computations have been F-cycles, with 2 sweeps before each prolongation and 2 sweeps before each restriction.

Time	single grid	2 grids	3 grids
1	22400	6526	1742
5	15200	4597	1319
10	15200	4547	1214
15	15200	4538	1214
20	15200	4536	1214

Table 5.2: Single-phase flow through an expansion - work units per time step for different number of grids

Time	single grid	2 grids	3 grids
1	62400	17338	4693
5	40800	11830	3119
10	40000	11830	3219
15	40000	11626	3219
20	40000	11626	3119

Table 5.3: Single-phase flow through a contraction - work units per time step for different number of grids

Time	single grid	2 grids	3 grids	4 grids	5 grids
1	1200	403	195	239	342
5	1000	354	167	244	347
10	1000	340	191	241	350
15	1000	341	193	249	350
20	1000	341	193	249	350

Table 5.4: Multi-phase channel flow - work units per time step for different number of grids

Time	single grid	2 grids	3 grids
1	7600	2158	878
5	6200	1893	791
10	6200	1894	780
15	6400	1894	785
20	6400	1894	785

Table 5.5: Multi-phase T-junction flow - work units per time step for different number of grids

Time	single grid	2 grids	3 grids
1	4000	1199	720
5	3200	1055	496
10	3200	1046	378
15	3200	1552	377
20	3200	999	385

Table 5.6: Multi-phase Contraction flow - work units per time step for different number of grids

5.5 Conclusions

In this Chapter, we have presented the results for the test cases selected to validate the `transient Pamg Single-phase` and `transient Pamg Multi-phase` solvers. Particular focus has been undertaken in the investigation of the accuracy and efficiency of the solvers. However, it was not in the aim of this thesis to be particularly exhaustive in this validation process. This is due to the fact that `transient Pamg Single-phase` and `transient Pamg Multi-phase` have been developed from the steady solvers `Pamg` and `Pamg multiphase`. Consequently, some behaviour of the transient solvers has been already observed in their steady state version.

The main conclusions are as follows:

- The implementation of the discretised equations for the single-phase and multi-phase regime (derived in Sections 3.2 and 3.3) have been validated by:
  - Choosing a set of test cases of varying complexity.
  - Doing a qualitative analysis of the solutions.
  - Verifying the grid independence (in space) of the solutions.
  - Checking the time step independence of the solutions.
  - Comparing the solutions with those provided by `CFX 4.3`.

  Since the results presented can be considered positive, it can be concluded that the implementations of the `transient Pamg Single-phase` and `transient Pamg Multi-phase` are correct.
- The use of automatic differentiation has been very important for computing in a very accurate way the Jacobians of the system of equations to solve. This has not been a surprise due to the non-linearity of the system of equations. Line-searching has been also necessary to guarantee the convergence of Newton's method.
- Very good multigrid acceleration has been observed. However, the convergence is not always grid independent, particularly for simpler problems. This must be due to the fact that multi-phase effects dominate the solution.
- The time-step independence of the solutions has been verified. This is due to the implicit method used for the formulation of the time dependent governing equations. However, for complex geometries like the contraction geometry, the chose of the time step has to be more careful. This is due to the fact that the velocities change more rapidly along the domain and consequently the jump of the time step has to be more cautious.

Chapter 6

Adaptivity. Coupled Solvers

6.1 Introduction

In this Chapter we are going to present and validate the new methodology for solving problems in the Computational Fluid Dynamics field for a particular set of test cases. This new methodology relies on the concept of dimensional adaptivity (D-adaptivity) which can be defined in the following manner. The dimension used by a solver in the simulation of a fluid flow depends on the geometry constraints of the problem and on the kinematic characteristics of the flows over the entire domain. In other words D-adaptivity consists of increasing or decreasing the number of variables used by a solver based on velocity criteria and on the geometry of the domain. In the scope of this Thesis we study the dimensional change between one and two space variables. However some of the processes established could be implemented in the dimensional change between one and three space variables.

During the course of this project several observations of the fluid flow behaviour in simple and complex geometries suggested the idea of the dimensional adaptivity. The simulation of a fluid flow through a contraction has been a very good example where this technique could be implemented. In fact the observation of the velocities over the entire domain suggests that there is some regions where a one-dimensional solver could simulate very well the fluid flow behaviour. Hence the more general solution given by the two-dimensional solver could be decomposed in some regions where a one-dimensional solution could be used.

Naturally an important point is how we decide the right moment for changing from a one-dimensional solver to the two-dimensional one and vice-versa. This depends on the way that we are going to change from the two solvers. If we are going to change from a one-dimensional solver to a two-dimensional one two triggers, which are related each other can be considered. One is the distance to the geometry constraints and the other the horizontal/axial velocity of the fluid flow. Strong evidence has been observed that suggest that the norm of the horizontal velocity must be lower than the horizontal distance to the geometry constraint. Naturally the dimensional change is only possible if the vertical/radial velocity is null or very close to zero. On the other hand if we are going to change from a

two-dimensional solver to a one-dimensional one the main trigger is the vertical velocity which must be very close to zero.

The implementation of this concept of D-adaptivity can be thought of in two different ways: The manual way and the automatic way. The manual way is based on the observation of the geometric constraints and velocity values produced by the two-dimensional solver over the entire domain. Using the rules outlined above the placement of the dimensional interfaces, i.e., where we are going to change from the one-dimensional solver to the two-dimensional one is decided. Naturally the automatic implementation of D-adaptivity is the implementation of the same rules but in a dynamic way such that the dimensional interfaces are decided as the simulation goes on. However an initial guess for the composition of the dimensional regions has to be outlined. This is based on the observation of the geometry and on the boundary and initial conditions of the problem.

The advantages of the D-adaptivity methodology outlined above are twofold. First of all we reduce drastically the computational cost on very complex problems. It is expected that using a one-dimensional solver in some regions of problems's domain will reduce substantially the CPU-time spent in a simulation. Secondly we allow a possible approach for computational fluid dynamics problems with very big domains. For instance to simulate a fluid flow through a very long pipe with a contraction the two-dimensional and three-dimensional solvers can not be applied to the entire domain.

Naturally such methodology has some disadvantages. One is some limitation on their use, which basically is due to some assumptions that are undertaken in the construction of the computational model. Another disadvantage is related to the accuracy of the solutions expected. Due to the mechanisms of changing from one-dimensional model to another dimensional model some errors are introduced, which necessarily have some effect on the accuracy of the overall solutions. Moreover, some assumptions made on the behaviour of the fluid flow through the overall domain may be not totally verified and consequently introduce another factor of growing errors.

Another important discussion is about the error control. This is related to the position where we decide to implement the interfaces. As the simulation of the fluid goes on, the decision to change from a one-dimensional region to the other dimensional one is based on the approximation of the solution that we have. Computing the error of the approximation and verifying if it is above a tolerance previously established does this. This error has to be computed with respect to the relevant components of the approximation. Naturally these are the components which are going to be lost when changing from dimensional regions. The measure of this error must be monitored in a neighbourhood of the interface to guarantee that no perturbations on the fluid flow will increase rapidly the error.

In the author's opinion the new ideas introduced during this project can be extrapolated to other fields. For instance the propagation of waves in radar could be thought of a dimensional coupling. In another field the heat transport through

a system of pipes could be another important application.

The scope of this thesis is laminar incompressible fluid flows. In the study that we carried out the fluid flow regime chosen has been the bubbly one. In the author's opinion the ideas outlined above can be also implemented in solvers prepared for slug and stratified flow regimes.

This Chapter is organised as follows.

First of all we introduce the treatment of the one-dimensional/two-dimensional interface domain for the numerical simulation of a single-phase incompressible flow. Navier-Stokes equations in Cartesian coordinates for one-dimensional flows are formulated. Integration over the interface region is discussed. Velocity interpolation formulas on the dimensional interfaces are deduced. This single-phase flow interface ends with discussing on the pressure drop issue.

Secondly, an asymptotic analysis on the steady well-mixed uni-directional flow is addressed. Starting from the one-dimensional steady multi-fluid model two alternative ways have been taken to derive the velocities and volume fractions expressions. One way based on the average resistance of the velocity components of the resistance and another way based on a standard asymptotic expansion with a small resistance coefficient. Expressions for the velocities and volume fractions are presented. Further discussion is addressed on this issue.

After having established the expressions for the velocities and volume fractions in cartesian coordinates these are validated against the computational results produced by the simulation of a two-phase fluid flow through a channel. Several experiments are used in this validation process and the behaviour of the velocities and volume fractions in each one is verified.

In Section 6.3.3 we discuss the implementation of the D-adaptivity on the construction of a `D-adaptive solver` based on the one-dimensional solver `Emaps` and on the `transient Pamg Multi-phase` which has been validated in Chapter 5. `Emaps` has been developed in the Applied Mathematics and Computing Group and has been validated by some papers (see for instance[78, 79]). The test case chosen has been the simulation of a two-phase fluid flow through a contraction. This problem due to its difficult geometry is necessarily a good example where the D-adaptivity can be employed. However an additional difficulty in the construction of this model has been raised. While `transient Pamg Multi-Phase` works with cartesian coordinates, `Emaps` has been designed to simulate fluid flows through pipes and consequently is prepared for cylindrical coordinates. Due to this difference between the results produced by the `D-adaptive solver` and the results produced by the `transient Pamg Multi-phase` over the entire domain are important.

In order to validate more accurately the D-adaptivity in Section 6.3.4 new asymptotic formulas in cylindrical coordinates have been derived. These have been implemented by the same techniques used in the Cartesian version. However, the expressions in cylindrical coordinates are much more complex and consequently this task necessarily has been harder.

Naturally after having derived the expressions for the velocities and volume fractions in cylindrical coordinates, the subsequent step was to proceed to its validation against computational results. This is done on Section 6.3.5. In this section results produced in an axial-symmetry pipe problem by `CFX 4.3` are compared with the analytic results produced by a function, which implement the velocities and volume fraction formulas derived. The characteristics of the fluids chosen in the experiments necessarily obey some assumptions that have been made in the asymptotic derivation process. However, the results observed in the test cases studied validate satisfactorily the analytic results.

Section 6.3.6 is where another implementation of the D-adaptivity is tested. In this case the `D-adaptive solver` constructed is based on `Emaps` and on the commercial code `CFX 4.3`. The idea was to construct a solver that worked in cylindrical coordinates and consequently could be adequately applied to the simulation of fluid flows through pipes. The geometry of the test problem implemented was a pipe with a contraction in the middle, which, in the author's opinion, is a characteristic of many flow problems and which occur several times in the Computational Fluid Dynamics field. Results are presented for the simulation of a two-phase laminar incompressible flow by the D-adaptive solver and by using the `CFX 4.3` solver over the whole domain. Good agreement has been achieved between the results produced by the two different methods.

6.2 One-Dimensional Two-Dimensional Coupling

6.2.1 Single-Phase Fluid Flow

In this section we are going to present the interpolation formulas for the interface region between one-dimensional/two-dimensional single-phase solvers and vice-versa. The starting point will be the formulation of the Navier-Stokes equations for a one-dimensional fluid flow in a channel. After establishing the velocity interpolation expression, we discuss the pressure drop along the one-dimensional domain.

On a one-dimensonal flow we have $v = 0$ so Equations (3.1), (3.2) and (3.3) become:

- Conservation of mass

$$\frac{\partial u}{\partial x} = 0 \tag{6.1}$$

- Conservation of horizontal momentum

$$\frac{\partial u}{\partial t} = -\frac{\partial p}{\partial x} + \nu \frac{\partial u^2}{\partial x^2} \tag{6.2}$$

- Conservation of vertical momentum

$$\frac{\partial p}{\partial y} = 0 \tag{6.3}$$

So we can approximate Equations (6.1), (6.2) and (6.3) for each time step by

$$\frac{\partial u}{\partial x} = 0 \tag{6.4}$$

$$\nu \frac{\partial u^2}{\partial x^2} = \frac{\partial p}{\partial x} \tag{6.5}$$

$$\frac{\partial p}{\partial y} = 0 \tag{6.6}$$

provided

$$\frac{\partial u}{\partial t} \ll 1$$

.

Solving Equation (6.5) in order of the unknown u we get

$$u(x,y) = \frac{1}{\nu}\frac{\partial p}{\partial x}\frac{y^2}{2} + By + C \tag{6.7}$$

In Figure 6.1 we represent the interface between a one-dimensional and a two-dimensional domain.

Let the initial conditions be $u = 0$, when $y = 0$ and $y = h$. Hence we get values for the integration constants and so Equation (6.7) becomes:

$$u(x,y) = \frac{1}{2\nu}\frac{\partial p}{\partial x} y(y-h) \tag{6.8}$$

Integrating in the domain:

$$\overline{u(x)} = \frac{1}{h}\int_0^h u dy = \frac{1}{h}\int_0^h \frac{1}{2\nu}\frac{\partial p}{\partial x} y(y-h) dy = \frac{1}{2h\nu}\frac{\partial p}{\partial x}\int_0^h y(y-h) dy \tag{6.9}$$

and finally get the equation:

$$\overline{u(x)} = \frac{h^2}{12\nu}\frac{\partial p}{\partial x} \tag{6.10}$$

From Equations (6.8) and (6.10) we get the interpolation formula:

$$u(x,y) = \frac{6\overline{u(x)} y(y-h)}{h^2} \tag{6.11}$$

Discretizing in the interface domain x_{itf1} we have

$$u(x_{itf1}, 1) = 0 \tag{6.12}$$

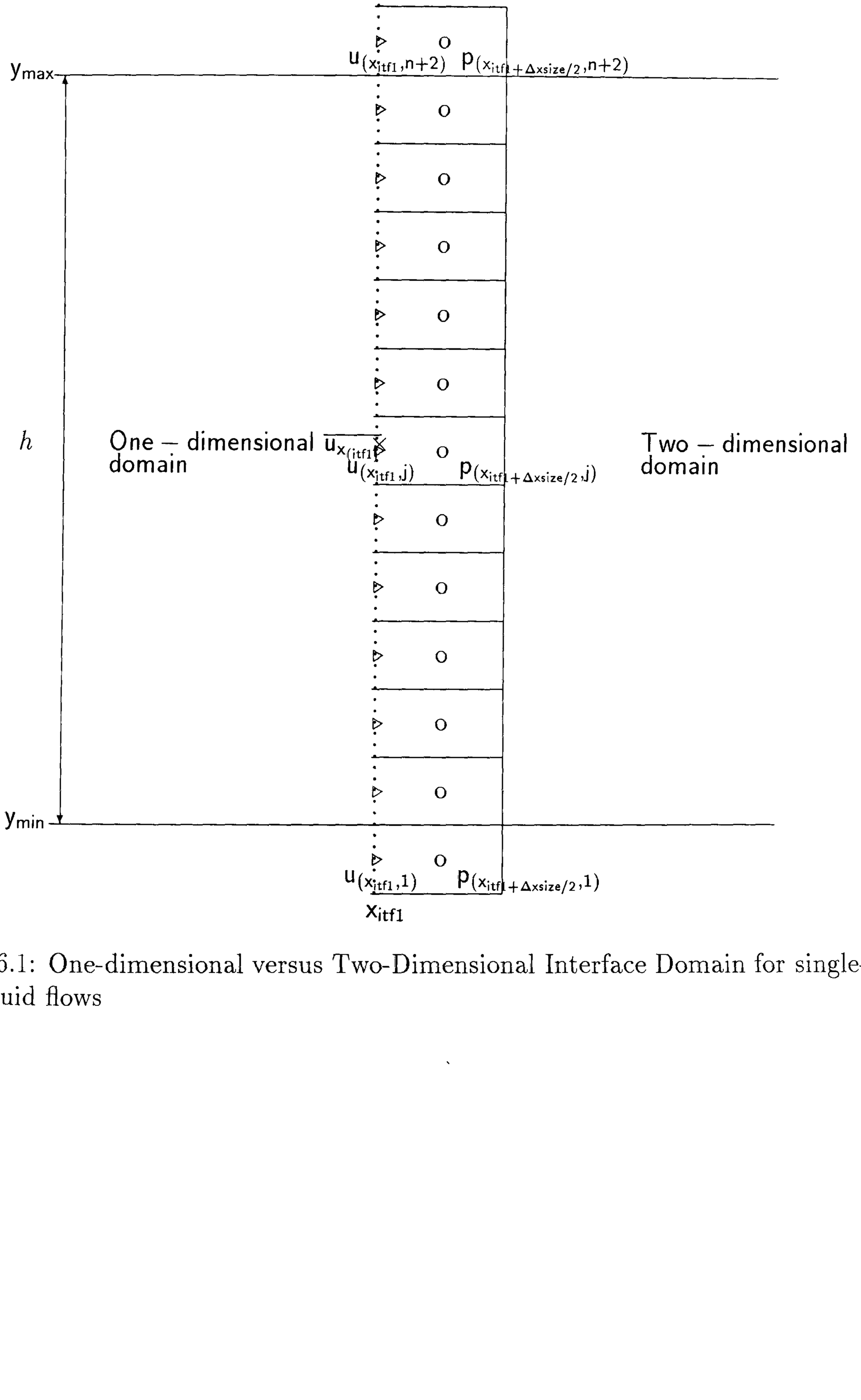


Figure 6.1: One-dimensional versus Two-Dimensional Interface Domain for single-phase fluid flows

$$u(x_{itf1}, j) = \frac{6\overline{u(x)}(\frac{(2j-3)\Delta y}{2} - h)(2j-3)}{h^2}\frac{\Delta y}{2}, \quad j = 2, ..., n+1 \tag{6.13}$$

$$u(x_{itf1}, n+2) = 0 \tag{6.14}$$

Δy is defined by

$$\Delta y = \frac{y_{max} - y_{min}}{n}$$

where y_{max} is the maximum ordinate, y_{min} is the minimum ordinate, n the number of cells and $h = y_{max} - y_{min}$.

Figure 6.2 represents the two-dimensional versus one-dimensional interface region.

In the two-dimensional/one-dimensional interface we will have the transformation:

$$\overline{u(x)} = \frac{1}{h}\lim_{n\to\infty}\sum_{j=2}^{n+1} u(x_{itf2}, j)\Delta y \approx \frac{1}{h}\sum_{j=2}^{n+1} u(x_{itf2}, j)\Delta y = \frac{1}{n}\sum_{j=2}^{n+1} u(x_{itf2}, j) \tag{6.15}$$

Pressure Drop

From Equation (6.3) we get the following equation:

$$p = f_1(x) \tag{6.16}$$

So, from Equations (6.5) and (6.8), we can conclude that the pressure only depends on x

$$\frac{\partial p}{\partial x} = k \Rightarrow p = k.x + C \tag{6.17}$$

We can determine k, solving Equation (6.10) in order of $\frac{\partial p}{\partial x}$:

$$\frac{\partial p}{\partial x} = \frac{12\nu\overline{u(x)}}{h^2} = k \tag{6.18}$$

Setting $p = 0$ when $x = 0$, we get

$$p = \frac{12\nu u(x)}{h^2}x$$

Hence the expression for the one-dimensional pressure, $\overline{p(x)}$ is

$$\overline{p(x)} = \frac{1}{h}\lim_{n\to\infty}\sum_{j=2}^{n+1} p(x, y_{min} + (2j-3)\frac{\Delta y}{2})\Delta y \approx \tag{6.19}$$

$$\frac{1}{n}\sum_{j=2}^{n+1} p(x, y_{min} + (2j-3)\frac{\Delta y}{2})$$

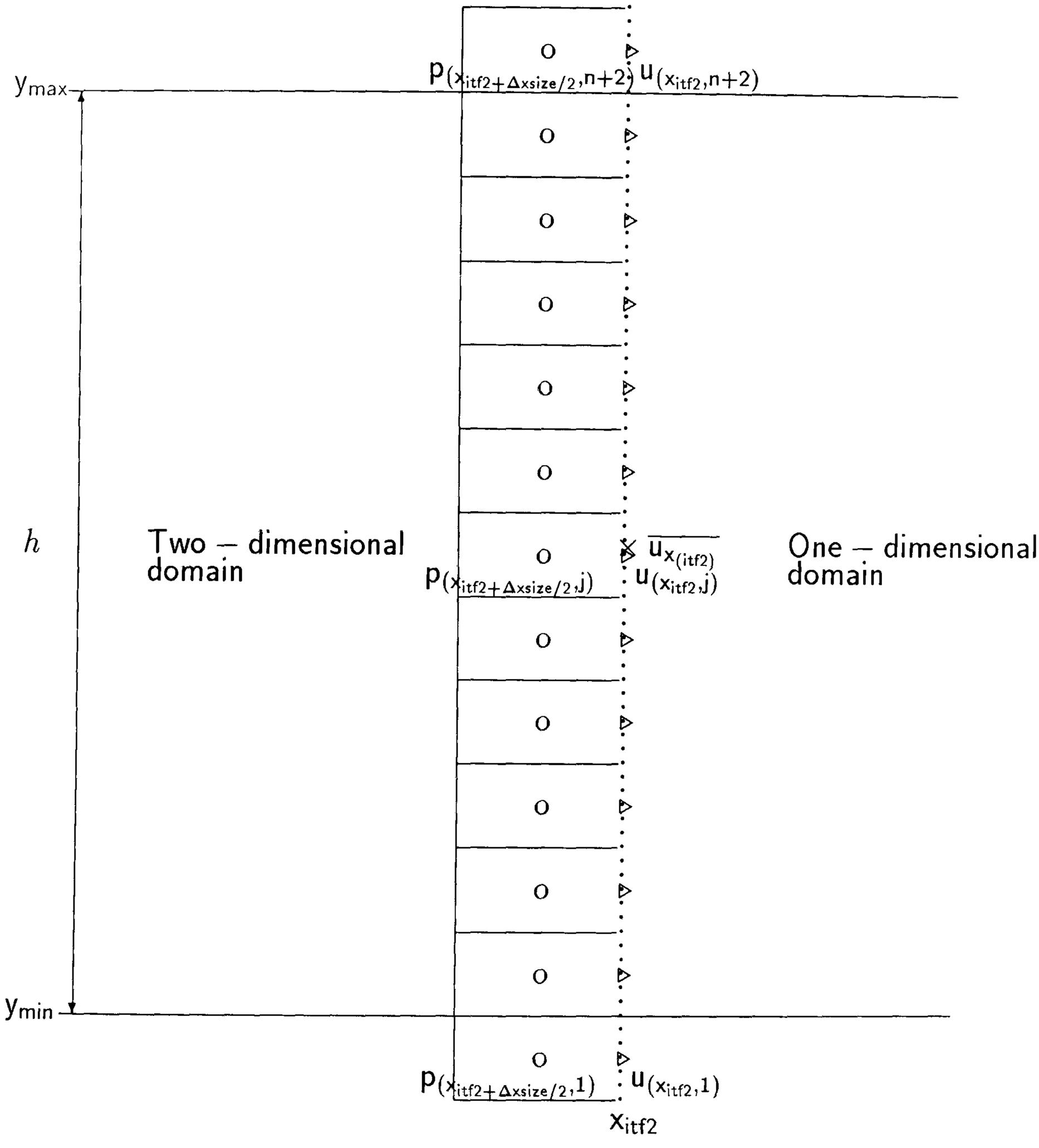


Figure 6.2: Two-dimensional versus One-Dimensional Interface Domain for single-phase fluid flows

In the line x_{itf1} we have

$$p(x, y_{min} + (2j-3)\frac{\Delta y}{2}) \approx \frac{\overline{p(x)}}{n} \tag{6.20}$$

where n is the number of cells In the line x_{itf2}, where we change from the one-dimensional domain to the two-dimensional one we get

$$\overline{p(x)} = \frac{1}{h}\int_0^h p(x,y)dy \approx \frac{1}{n}\sum_{j=2}^{n+1} p(x, y_{min} + (2j-3)\frac{\Delta y}{2}) \tag{6.21}$$

In this section we derived the new interpolation formulas for the coupling of a one-dimensional/two-dimensional single-phase solvers. Navier-Stokes equations for a one-dimensional fluid flow in a channel have been presented. Expressions for the interpolation of the velocities in the one-dimensional/two-dimensional interfaces have been derived. Similarly expressions for the interpolation of the pressure values in the same interfaces have been discussed.

Since our main objective is the coupling of multi-phase solvers no additional research has been done with single-phase solvers. Naturally it is expected that the formulation of the multi-phase interpolation formulas will cover the single-phase case. This relies on the fact that single-phase solvers can be seen as a particular case of the multi-phase solvers.

6.3 Two-Phase Fluid Flow

6.3.1 Asymptotic analysis of well-mixed two-phase flow in cartesian coordinates

In Section 6.2.1 we have outlined the treatment of the interface region between one-dimensional/two-dimensional single-phase solvers.

In this section we are going to discuss the formulas to apply when changing from a one-dimensional domain to a two-dimensional domain when simulating multi-phase fluid flows through a channel.

In order to achieve our goals we are going to present an asymptotic analysis of well-mixed two-phase flow[80]. For a general book about asymptotic treatment of differential equations see for instance reference[81]. The main objective that we are looking for is to establish the correct formulas for velocities and volume fractions that are going to be used in the one-dimensional/two-dimensional interface region for multi-phase fluid flows.

The starting point for this approach are the steady state version of the multi-fluid differential equations (3.54), (3.56) and (3.57) derived in Section 3.3. Since the fluid is unidirectional, $v_\alpha = 0$ and for the simulation of a flow through a long pipe it is expected that $\frac{\partial u_\alpha}{\partial x}$ will be null.

Hence the governing equations for a steady unidirectional two-phase fluid flow are.

$$\mu_\alpha \frac{\partial}{\partial y}\left(r_\alpha \frac{\partial u_\alpha}{\partial y}\right) = r_\alpha \frac{\partial p}{\partial x} - \frac{C_D}{d_{\alpha\beta}} r_\alpha r_\beta (r_\alpha \rho_\alpha + r_\beta \rho_\beta)(u_\beta - u_\alpha)|u_\beta - u_\alpha| \quad (6.22)$$

$$\mu_\beta \frac{\partial}{\partial y}\left(r_\beta \frac{\partial u_\beta}{\partial y}\right) = r_\beta \frac{\partial p}{\partial x} - \frac{C_D}{d_{\alpha\beta}} r_\alpha r_\beta (r_\alpha \rho_\alpha + r_\beta \rho_\beta)(u_\alpha - u_\beta)|u_\alpha - u_\beta| \; . \quad (6.23)$$

Scaling velocity with U, pressure with P and y with H the problem may be written in non-dimensional form

$$\frac{\partial}{\partial y}\left(r_\alpha \frac{\partial u_\alpha}{\partial y}\right) = r_\alpha \frac{\partial p}{\partial x} + f\lambda(u_\alpha - u_\beta)|u_\alpha - u_\beta| \quad (6.24)$$

$$\frac{\partial}{\partial y}\left(r_\beta \frac{\partial u_\beta}{\partial y}\right) = \frac{\mu_\alpha}{\mu_\beta}\left(r_\beta \frac{\partial p}{\partial x} - f\lambda(u_\alpha - u_\beta)|u_\alpha - u_\beta|\right) \; , \quad (6.25)$$

where the pressure scale has been chosen as $P = \mu_\alpha U L/H^2$ and

$$\lambda = \frac{C_D U H^2}{\mu_\alpha d_{\alpha\beta}} \qquad f = r_\alpha r_\beta (r_\alpha \rho_\alpha + r_\beta \rho_\beta) \; .$$

The governing equation for flow in the y direction indicates $p = p(x)$. The system is closed with the following relation :

$$r_\alpha + r_\beta = 1 \quad (6.26)$$

$$Q_\alpha = r_\alpha \int_0^1 u_\alpha \, dy \quad (6.27)$$

$$Q_\beta = r_\beta \int_0^1 u_\beta \, dy \; . \quad (6.28)$$

In the following analysis this system will be tackled in two ways. In the first an average value of the resistance term $(u_\alpha - u_\beta)|u_\alpha - u_\beta| \sim (Q_\alpha/r_\alpha - Q_\beta/r_\beta)|Q_\alpha/r_\alpha - Q_\beta/r_\beta|$ will be used. In the second a standard asymptotic expansion based on a small resistance coefficient λ will be taken.

Averaged resistance

The problem is now governed by

$$\frac{\partial}{\partial y}\left(r_\alpha \frac{\partial u_\alpha}{\partial y}\right) = r_\alpha \frac{\partial p}{\partial x} + f\lambda\Gamma \quad (6.29)$$

$$\frac{\partial}{\partial y}\left(r_\beta \frac{\partial u_\beta}{\partial y}\right) = \frac{\mu_\alpha}{\mu_\beta}\left(r_\beta \frac{\partial p}{\partial x} - f\lambda\Gamma\right) \; , \quad (6.30)$$

where $\Gamma = (Q_\alpha/r_\alpha - Q_\beta/r_\beta)|Q_\alpha/r_\alpha - Q_\beta/r_\beta|$ and the non-dimensional flux $Q = Q'/(UH)$. Since Γ is constant these equations may be integrated immediately

$$r_\alpha u_\alpha = \frac{1}{2}\left(r_\alpha \frac{\partial p}{\partial x} + f\lambda\Gamma\right) y(y-1) \quad (6.31)$$

$$r_\beta u_\beta = \frac{1}{2}\frac{\mu_\alpha}{\mu_\beta}\left(r_\beta \frac{\partial p}{\partial x} - f\lambda\Gamma\right) y(y-1) \; . \quad (6.32)$$

The fluxes are

$$Q_\alpha = -\frac{1}{12}\left(r_\alpha \frac{\partial p}{\partial x} + f\lambda\Gamma\right) \tag{6.33}$$

$$Q_\beta = -\frac{1}{12}\frac{\mu_\alpha}{\mu_\beta}\left(r_\beta \frac{\partial p}{\partial x} - f\lambda\Gamma\right) . \tag{6.34}$$

This allows a simpler expression for the velocities

$$u_\alpha = -\frac{6Q_\alpha}{r_\alpha} y(y-1) \tag{6.35}$$

$$u_\beta = -\frac{6Q_\beta}{r_\beta} y(y-1) . \tag{6.36}$$

Eliminating the pressure gradient gives an equation involving r_α, r_β

$$12\left(\mu_\alpha Q_\alpha r_\beta - \mu_\beta Q_\beta r_\alpha\right) + \mu_\alpha f\lambda\Gamma\left(r_\alpha + r_\beta\right) = 0 . \tag{6.37}$$

Finally, we can get the following equation for r_α

$$\begin{aligned} 12 r_\alpha r_\beta \left(\mu_\alpha Q_\alpha r_\beta - \mu_\beta Q_\beta r_\alpha\right) \; + \\ \mu_\alpha \lambda \left(r_\alpha + r_\beta\right)\left(r_\alpha \rho_\alpha + r_\beta \rho_\beta\right)\left(r_\beta Q_\alpha - r_\alpha Q_\beta\right)\left|r_\beta Q_\alpha - r_\alpha Q_\beta\right| = 0 . \end{aligned} \tag{6.38}$$

This has only one solution r_α so that $0 \le r_\alpha \le 1$.

Small λ expansion

A more general approach is to consider a expansion of a small parameter λ. Assume $\lambda \ll 1$ and look for solutions of the form

$$u = u_0 + \lambda u_1 + \cdots \tag{6.39}$$

$$r = r_0 + \lambda r_1 + \cdots . \tag{6.40}$$

The leading order problem is governed by

$$\frac{\partial}{\partial y}\left(r_{\alpha 0}\frac{\partial u_{\alpha 0}}{\partial y}\right) = r_{\alpha 0}\frac{\partial p}{\partial x} \tag{6.41}$$

$$\frac{\partial}{\partial y}\left(r_{\beta 0}\frac{\partial u_{\beta 0}}{\partial y}\right) = \frac{\mu_\alpha}{\mu_\beta} r_{\beta 0}\frac{\partial p}{\partial x} \tag{6.42}$$

$$r_{\alpha 0} + r_{\beta 0} = 1 . \tag{6.43}$$

As the flow is unidirectional and incompressible, the continuity equation requires v is constant. Since $v = 0$ at the walls $v = v_0 = v_1 = \cdots = 0$ and there will be no vertical motion. In which case, the volume fractions r_i must be constant at all orders.

The leading order velocities are therefore

$$u_{\alpha 0} = \frac{1}{2}\frac{\partial p}{\partial x} y(y-1) \tag{6.44}$$

$$u_{\beta 0} = \frac{\mu_\alpha}{2\mu_\beta}\frac{\partial p}{\partial x} y(y-1) \ . \tag{6.45}$$

The fluxes are specified by equations (6.27) and (6.28). These allow expressions to be obtained for the velocities in terms of the flux

$$u_{\alpha 0} = -\frac{6Q_\alpha}{r_{\alpha 0}} y(y-1) \tag{6.46}$$

$$u_{\beta 0} = -\frac{6Q_\beta}{r_{\beta 0}} y(y-1) \ . \tag{6.47}$$

Note, these are exactly the same as the expressions obtained via the average process of the previous section. The averaging method therefore provides the correct leading order solution for small interphase mixing.

Equating expressions for the pressure gradient obtained via (6.27) and (6.28) leads to

$$\mu_\alpha Q_\alpha r_{\beta 0} = \mu_\beta Q_\beta r_{\alpha 0} \ . \tag{6.48}$$

This corresponds to equation (6.39) with $\lambda = 0$. Combining this with equation (6.26) gives

$$r_{\alpha 0} = \frac{\mu_\alpha Q_\alpha}{\mu_\alpha Q_\alpha + \mu_\beta Q_\beta} \qquad r_{\beta 0} = \frac{\mu_\beta Q_\beta}{\mu_\alpha Q_\alpha + \mu_\beta Q_\beta} \ . \tag{6.49}$$

The leading order problem is therefore completely solved, with the velocities and volume fractions given by equations (6.46), (6.47) and (6.49). The velocity expressions are identical to those obtained by averaging. However, in this case the leading order volume fractions can be expressed simply in terms of the viscosities and fluxes. The result is different from that obtained by averaging.

The first order problem is governed by

$$\frac{\partial}{\partial y}\left(r_{\alpha 0}\frac{\partial u_{\alpha 1}}{\partial y} + r_{\alpha 1}\frac{\partial u_{\alpha 0}}{\partial y}\right) = r_{\alpha 1}\frac{\partial p}{\partial x} + g_0 \tag{6.50}$$

$$\frac{\partial}{\partial y}\left(r_{\beta 0}\frac{\partial u_{\beta 1}}{\partial y} + r_{\beta 1}\frac{\partial u_{\beta 0}}{\partial y}\right) = \frac{\mu_\alpha}{\mu_\beta}\left(r_{\beta 1}\frac{\partial p}{\partial x} - g_0\right) \tag{6.51}$$

$$r_{\alpha 1} + r_{\beta 1} = 0 \ , \tag{6.52}$$

where

$$g_0 = r_{\alpha 0} r_{\beta 0}(r_{\alpha 0}\rho_\alpha + r_{\beta 0}\rho_\beta)\,(u_{\alpha 0} - u_{\beta 0})\,|u_{\alpha 0} - u_{\beta 0}| \ . \tag{6.53}$$

From Equations (6.44) and (6.45) we get

$$\mu_\alpha u_{\alpha 0} = \mu_\beta u_{\beta 0} \tag{6.54}$$

Eliminating g_0 from equations (6.50), (6.51), applying (6.52) and also using Equation (6.54) we get

$$\frac{\partial}{\partial y}\left(r_{\alpha 0}\frac{\partial u_{\alpha 1}}{\partial y}+\frac{\mu_\beta}{\mu_\alpha}r_{\beta 0}\frac{\partial u_{\beta 1}}{\partial y}\right)=0\ . \tag{6.55}$$

This may be integrated twice, subject to $u_{\alpha 1}=u_{\beta 1}=0$ at $y=0,\ 1$ to give

$$r_{\alpha 0}u_{\alpha 1}=-\frac{\mu_\beta}{\mu_\alpha}r_{\beta 0}u_{\beta 1}\ . \tag{6.56}$$

Subtracting equation (6.51) multiplied by $\frac{\mu_\beta}{\mu_\alpha}$ from (6.50) gives

$$\frac{\partial}{\partial y}\left(r_{\alpha 0}\frac{\partial u_{\alpha 1}}{\partial y}+r_{\alpha 1}\frac{\partial u_{\alpha 0}}{\partial y}-\frac{\mu_\beta}{\mu_\alpha}\left(r_{\beta 0}\frac{\partial u_{\beta 1}}{\partial y}+r_{\beta 1}\frac{\partial u_{\beta 0}}{\partial y}\right)\right) \;=\; 2\left(r_{\alpha 1}\frac{\partial p}{\partial x}+g_0\right)\ . \tag{6.57}$$

This may be simplified using (6.56) and (6.52)

$$\frac{\partial}{\partial y}\left(r_{\alpha 0}\frac{\partial u_{\alpha 1}}{\partial y}+r_{\alpha 1}\frac{\partial u_{\alpha 0}}{\partial y}\right) \;=\; r_{\alpha 1}\frac{\partial p}{\partial x}+g_0\ . \tag{6.58}$$

Before integrating this equation the function g_0 may be expressed as

$$g_0 \;=\; r_{\alpha 0}r_{\beta 0}(r_{\alpha 0}\rho_\alpha+r_{\beta 0}\rho_\beta)36y^2(y-1)^2\left(\frac{Q_\beta}{r_{\beta 0}}-\frac{Q_\alpha}{r_{\alpha 0}}\right)\left|\frac{Q_\beta}{r_{\beta 0}}-\frac{Q_\alpha}{r_{\alpha 0}}\right| \tag{6.59}$$

$$\;=\; -Ay^2(1-y)^2\ , \tag{6.60}$$

where A is constant.

If $r_{\alpha 1}$ is allowed to be an arbitrary function of y then (6.58) must be integrated numerically, subject to

$$u_{\alpha 1}|_{y=0}\ u_{\alpha 1}|_{y=1}=0 \tag{6.61}$$

$$\int_0^1 r_{\alpha 1}u_{\alpha 1}\ dy=0 \tag{6.62}$$

However, due to the form of previous solutions it seems worth trying

$$r_{\alpha 1}=a_1y(y-1)\ . \tag{6.63}$$

Equation (6.58) may be integrated analytically. Applying (6.62) and the no-slip conditions leads to

$$a_1=-\frac{11A}{24}\left(\frac{\partial p}{\partial x}\right)^{-1}\ . \tag{6.64}$$

The velocity is

$$r_{\alpha 0}u_{\alpha 1}=-\frac{Ay}{1440}\left(48y^5-144y^4+10y^3+220y^2-165y+31\right)\ . \tag{6.65}$$

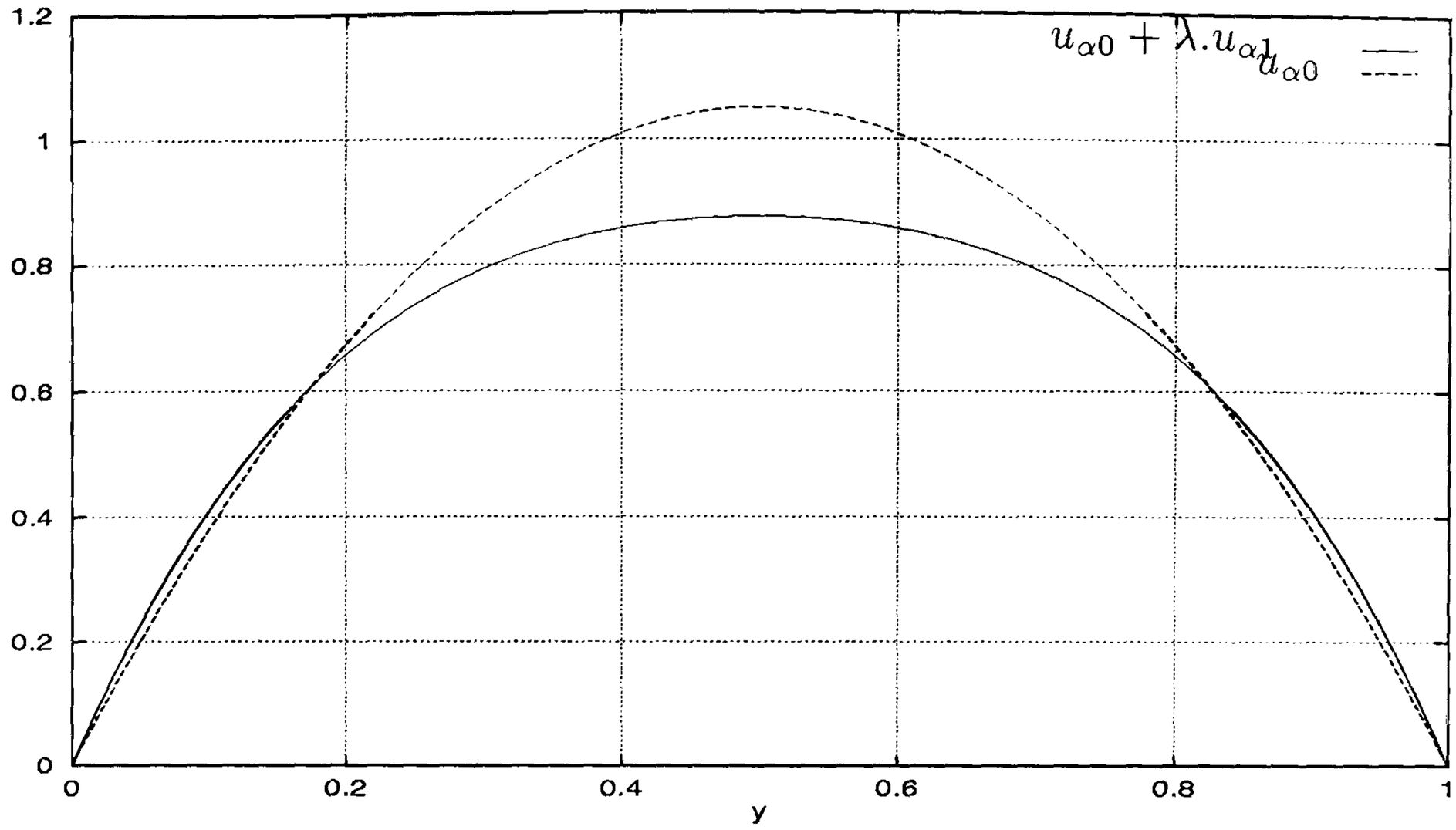


Figure 6.3: Comparison of u_α to leading and first order. It can be seen that over the central region the faster fluid is slowed down by the resistance. At the edges it speeds up slightly.

Equation (6.56) can be used to obtain an expression for $u_{\beta 1}$.

Figure 6.3 shows a plot of $u_{\alpha 0}$ and $u_{\alpha 0} + \lambda u_{\alpha 1}$. Figure 6.4 shows a plot of $u_{\beta 0}$ and $u_{\beta 0} + \lambda u_{\beta 1}$. In Figure 6.3 it can be seen that over the central region the faster fluid is slowed down by the resistance. At the edges it speeds up slightly. Similarly, the slower fluid shown in Figure 6.4 speeds in the central region but slows down slightly near the edges.

In dimensional form we get the following expressions for the leading order velocities:

$$u_{\alpha 0}^{'} = \frac{1}{2\mu_\alpha} \frac{\partial p^{'}}{\partial x^{'}} y^{'}(y^{'} - H) \tag{6.66}$$

$$u_{\beta 0}^{'} = \frac{1}{2\mu_\beta} \frac{\partial p^{'}}{\partial x^{'}} y^{'}(y^{'} - H)\ . \tag{6.67}$$

where

$$u_{\alpha 0}^{'} = U u_{\alpha 0}$$

$$u_{\beta 0}^{'} = U u_{\beta 0}$$

$$p^{'} = P p$$

$$x^{'} = L x$$

$$y^{'} = H y$$

$$P = \frac{\mu_\alpha U L}{H^2}$$

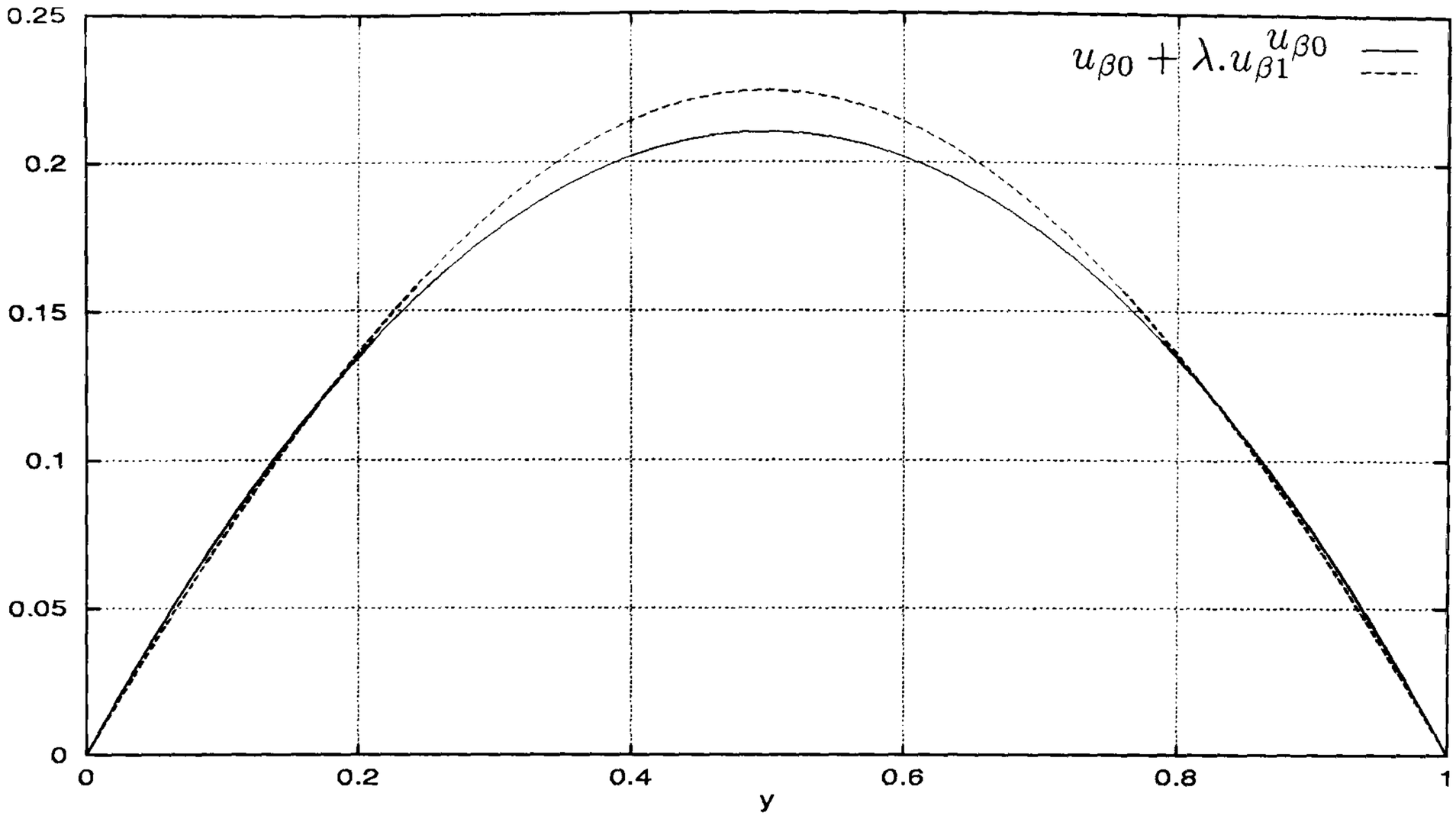


Figure 6.4: Comparison of u_β to leading and first order. The slower fluid speeds in the central region but slows down slightly near the edges.

So the dimensional form of $u_{\alpha 0} + \lambda u_{\alpha 1}$ and $u_{\beta 0} + \lambda u_{\beta 1}$ are the expressions:

$$u'_{\alpha 0} + \lambda u'_{\alpha 1} = \frac{1}{2\mu_\alpha}\frac{\partial p'}{\partial x'}y'(y' - H) - \frac{36 C_D r_{\beta 0}(r_{\alpha 0}\rho_\alpha + r_{\beta 0}\rho_\beta)}{1440\mu_\alpha d_{\alpha\beta}}\left(\frac{Q'_\beta}{r_{\beta 0}} - \frac{Q'_\alpha}{r_{\alpha 0}}\right) \mid \frac{Q'_\beta}{r_{\beta 0}} - \frac{Q'_\alpha}{r_{\alpha 0}} \mid$$
$$y'\left(48\frac{y'^5}{H^6} - 144\frac{y'^4}{H^5} + 10\frac{y'^3}{H^4} + 220\frac{y'^2}{H^3} - 165\frac{y'}{H^2} + \frac{31}{H}\right) \tag{6.68}$$

$$u'_{\beta 0} + \lambda u'_{\beta 1} = \frac{1}{2\mu_\beta}\frac{\partial p'}{\partial x'}y'(y' - H) - \frac{36 C_D r_{\alpha 0}(r_{\alpha 0}\rho_\alpha + r_{\beta 0}\rho_\beta)}{1440\mu_\beta d_{\alpha\beta}}\left(\frac{Q'_\beta}{r_{\beta 0}} - \frac{Q'_\alpha}{r_{\alpha 0}}\right) \mid \frac{Q'_\beta}{r_{\beta 0}} - \frac{Q'_\alpha}{r_{\alpha 0}} \mid$$
$$y'\left(48\frac{y'^5}{H^6} - 144\frac{y'^4}{H^5} + 10\frac{y'^3}{H^4} + 220\frac{y'^2}{H^3} - 165\frac{y'}{H^2} + \frac{31}{H}\right) \tag{6.69}$$

where

$$Q'_\alpha = UHQ_\alpha$$
$$Q'_\beta = UHQ_\beta$$

and

$$\lambda = \frac{C_D U H^2}{\mu_\alpha d_{\alpha\beta}}.$$

The expressions for the volume fractions are

$$r_{\alpha 0} + \lambda r_{\alpha 1} = \frac{\mu_\alpha Q'_\alpha}{\mu_\alpha Q'_\alpha + \mu_\beta Q'_\beta} - \frac{11}{96}\frac{C_D}{d_{\alpha\beta}} r_{\alpha 0} r_{\beta 0}(r_{\alpha 0}\rho_\alpha + r_{\beta 0}\rho_\beta)$$

$$H^6 \frac{\partial p^{'}}{\partial x^{'}} (\frac{1}{\mu_\alpha} - \frac{1}{\mu_\beta}) | \frac{1}{\mu_\alpha} - \frac{1}{\mu_\beta} | y^{'}(H - y^{'}) \tag{6.70}$$

and

$$r_{\beta 0} + \lambda r_{\beta 1} = \frac{\mu_\beta Q^{'}_\beta}{\mu_\alpha Q^{'}_\alpha + \mu_\beta Q^{'}_\beta} + \frac{11}{96} \frac{C_D}{d_{\alpha\beta}} r_{\alpha 0} r_{\beta 0} (r_{\alpha 0}\rho_\alpha + r_{\beta 0}\rho_\beta)$$
$$H^6 \frac{\partial p^{'}}{\partial x^{'}} (\frac{1}{\mu_\alpha} - \frac{1}{\mu_\beta}) | \frac{1}{\mu_\alpha} - \frac{1}{\mu_\beta} | y^{'}(H - y^{'}) \tag{6.71}$$

In this section we have established the new expressions for the velocities and volume fractions to use when changing from a one-dimensional domain to a two-dimensional one. Establishing an asymptotic analysis of well-mixed two-phase flow has done this. Some assumptions have been made which in certain sense will limit the range of application of these formulas. One of these assumptions is that the horizontal velocity does not depend on the horizontal distance. This is not always true but we expect that for a long pipe it will happen. Another assumption is that there will be no vertical mixing since the flow is unidirectional. This is true if the fluid flows are not very different in their densities. Otherwise it is expected that there will be important vertical mixing and consequently the volume fractions will not be constant. Naturally another condition that limit the validation of the formulas is the size of the parameter used in the expansion. This parameter has to be very small otherwise the truncation error due to the approximation of the velocities with the leading order terms will be very big.

In spite of the assumptions that have been undertaken the asymptotic formulas for the velocities and volume fractions can be very useful. They can be used in the interface treatment of the one-dimensional/two-dimensional two-phase solvers. Naturally the fluid flows and the geometry of the problems have to satisfy the assumptions outlined above. Before using this asymptotic formulas in the coupling of a one-dimensional solver with a two-dimensional one we are going to validate them with the results produced by `transient Pamg Multi-phase` in a known problem. This is precisely the objective of next section.

6.3.2 Validation in a Two-Phase Flow through a Channel

In this section we are going to validate the asymptotic formulas derived for the steady two-phase fluid flow (see Section 6.3.1). This process is established by the comparison of the computational results with the analytical solution of two-phase fluid flow in a channel. A set of fluids has been used for this purpose. The main differences between each one are in the viscosities. Furthermore, since the geometry and the boundary conditions used are always identical, the Reynolds numbers for each test case are different because of the viscosities chosen.

The geometry for this test case is exactly the same used in the simulation of the single-phase fluid flow through a channel (see Chapter 5 Figure 5.1). The boundary conditions used are at the inlet, Dirichlet boundary conditions and at the outlet Neumann conditions. In the inlet a parabolic velocity profile is specified for the x-component velocity $u(x, y)$ by

$$u(0, y) = 4y(1 - y)$$

and for the y-component velocity $v(x, y)$ it is null in the boundary, i.e. $v(0, y) = 0$. The volume fractions in the inlet are set to 0.5 for both phases. At the outlet the vertical velocity components for both phases and the volume fractions are null.

The values chosen for the inter-facial length are $d_{\alpha\beta} = d_{\beta\alpha} = 0.1$. For the drag coefficient three different values have been used: $C_D = 0.01$, $C_D = 0.001$ and $C_D = 0.0001$. Hence the corresponding parameter $\lambda = C_D/d_{\alpha\beta}$ satisfy $\lambda \ll 1$ since it takes the values 0.1, 0.01 and 0.001.

The physical properties chosen for the first test case are:

Two-Phase Flow	Reynolds Number	Viscosity	Density
Phase α	100	0.01	1.0
Phase β	200	0.005	1.0

Figures 6.5, 6.6 and 6.7 present the comparison between the velocity profiles computed by the `transient Pamg Multi-phase` solver and the velocity computed analytically by Equations (6.68) and (6.69) at the outlet ($x = 3.0$).

It is visible that when λ decreases the curves which represent the computational velocities agree well with the analytically calculated ones. This is precisely what we are expecting since the asymptotic formulas derived in Section 6.3.1 will be more accurate when the parameter λ used in the velocities expansion becomes smaller. Furthermore, we can conclude that as λ becomes smaller, and consequently very close to zero, the relation of the velocities of the two phases α and β is 1 to 2. In fact, when $\lambda = 0.0$ Equations (6.68) and (6.69) will be reduced to the expressions of the leading order velocities

$$u_{\alpha 0}^{'} = \frac{1}{2\mu_\alpha}\frac{\partial p^{'}}{\partial x^{'}} y^{'}(y^{'} - H)$$

$$u_{\beta 0}^{'} = \frac{1}{2\mu_\beta}\frac{\partial p^{'}}{\partial x^{'}} y^{'}(y^{'} - H)\ .$$

and consequently $u_{\beta 0}^{'} = \frac{\mu_\alpha}{\mu_\beta} u_{\alpha 0}^{'}$ which for the viscosities of our test case lead to $u_{\beta 0}^{'} = 2u_{\alpha 0'}$.

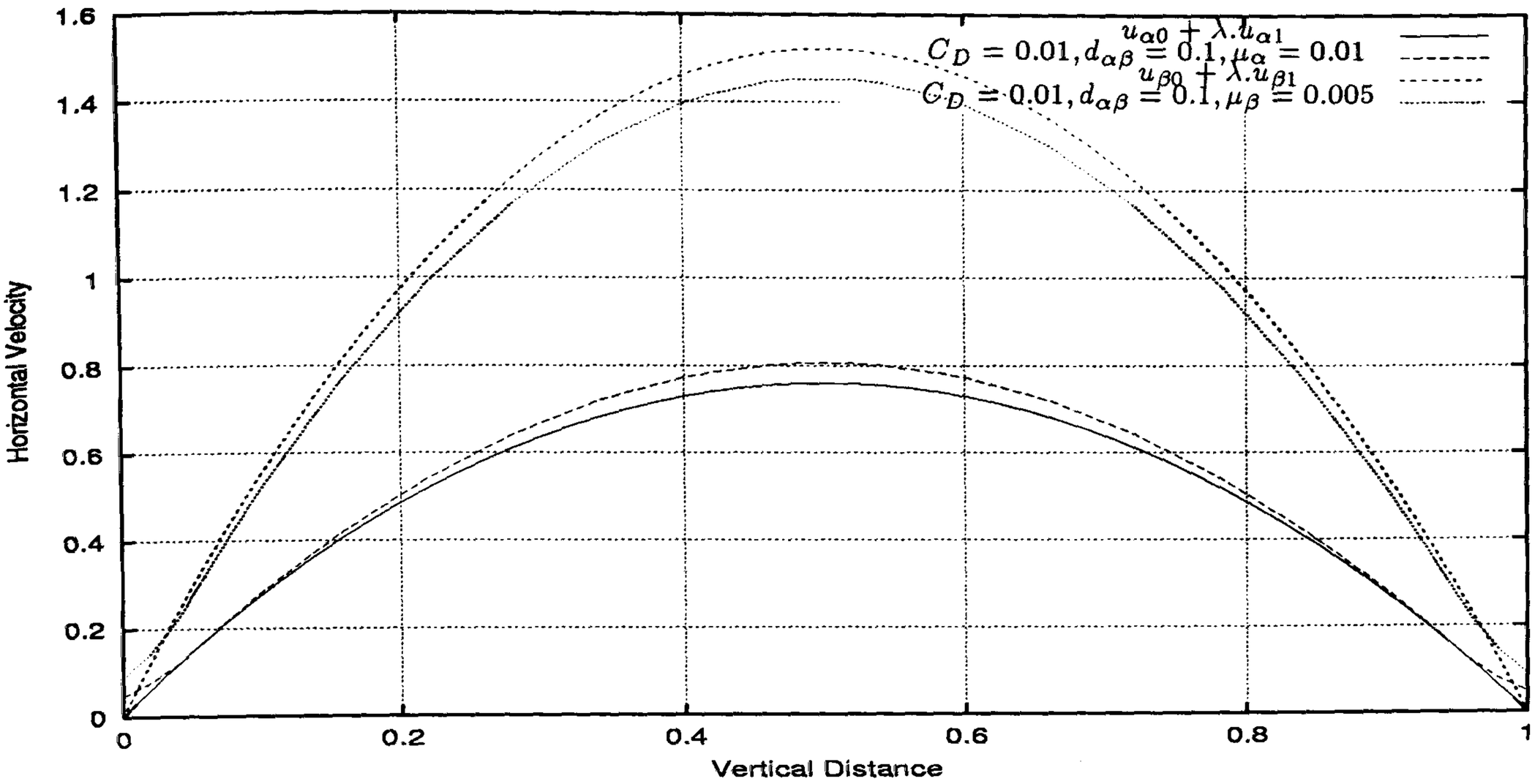


Figure 6.5: Two-phase channel flow - comparison between the analytical horizontal velocity and the computational one along the line $x = 3.0$ for $\lambda = 0.1$. It can be seen the good agreement between the computational results and the analytical ones. The slight jumps observed in the computational results at the walls are due to interpolation errors.

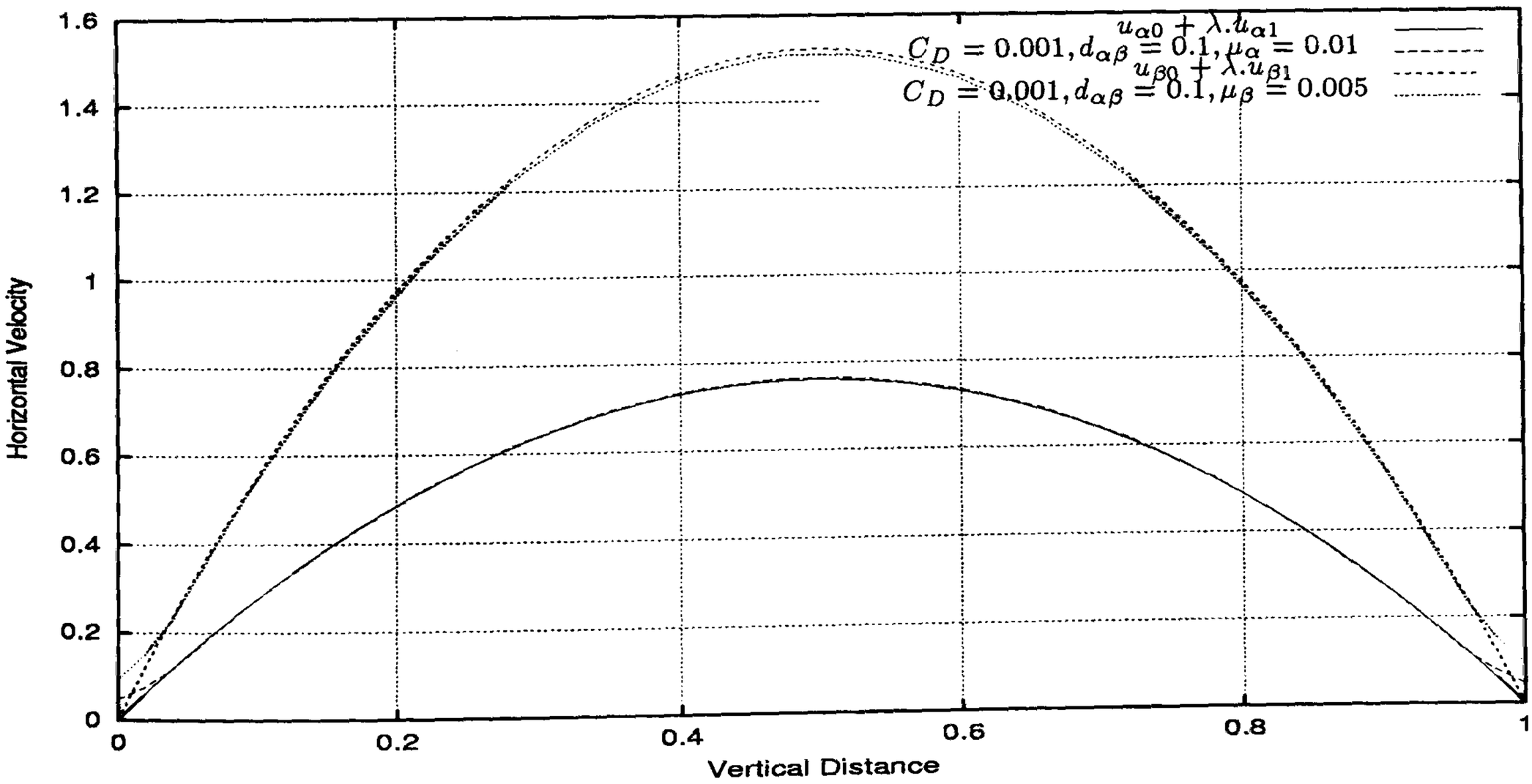


Figure 6.6: Two-phase channel flow - comparison between the analytical horizontal velocity and the computational one along the line $x = 3.0$ for $\lambda = 0.01$. Good agreement between the analytical results and the computational ones can be observed.

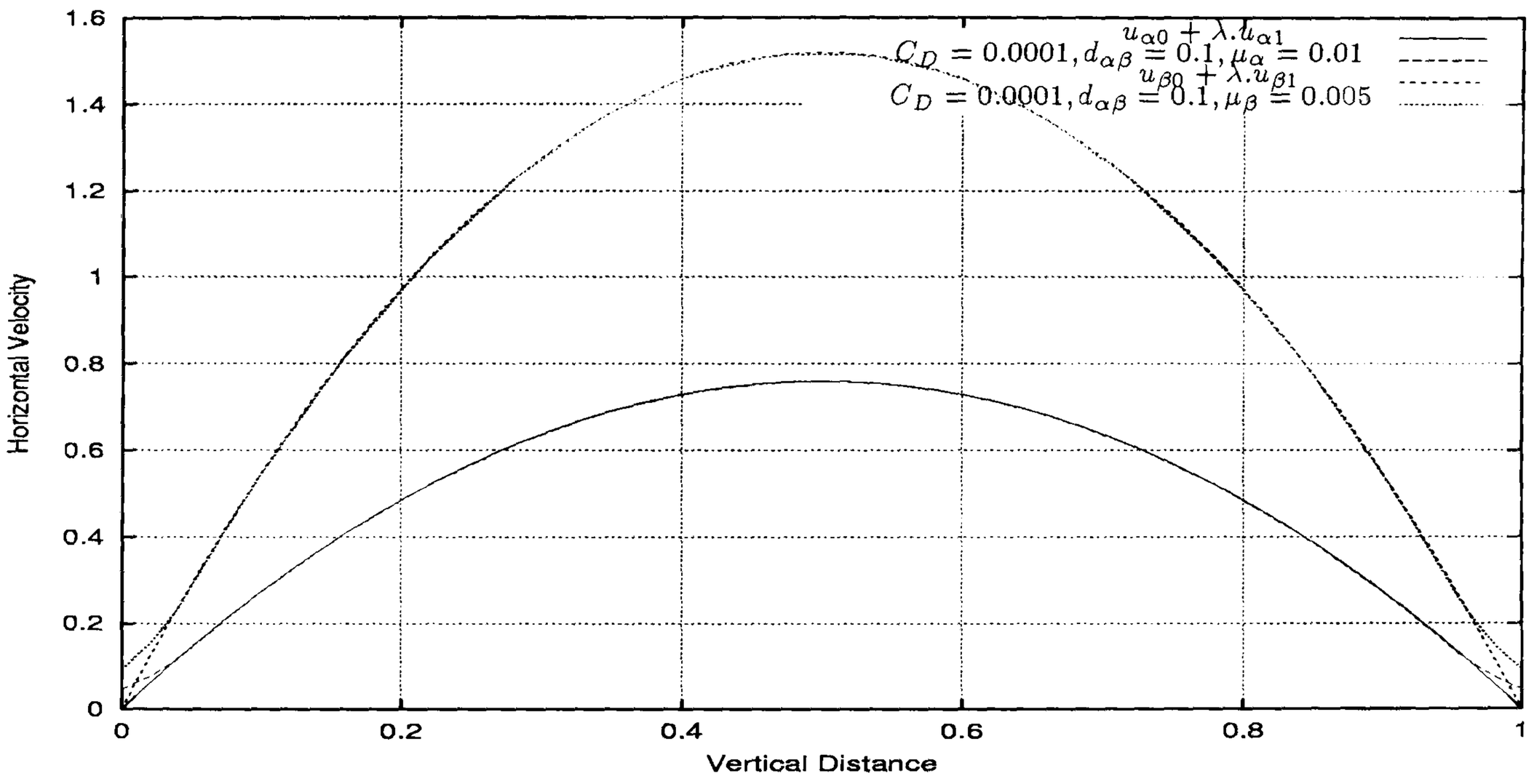


Figure 6.7: Two-phase channel flow - comparison between the analytical horizontal velocity and the computational one along the line $x = 3.0$ for $\lambda = 0.001$. It can be seen the excellent agreement between the computational results and the analytical ones.

Another test case has been outlined with the same domain, boundary conditions and values of λ. The only difference relies on the viscosity of phase β. Consequently the relation between the viscosities of the two-phases α and β is different than the example above. The physical properties chosen for this second test case are resumed in the table:

Two-Phase Flow	Reynolds Number	Viscosity	Density
Phase α	100	0.01	1.0
Phase β	10	0.1	1.0

In Figures 6.8, 6.9 and 6.10 we see the velocity profiles produced by the `transient Pamg Multi-phase` solver and the analytic Equations 6.68 and 6.69 at the outlet ($x = 3.0$).

In this test case the behaviour observed in the predecessor case is still verified. In fact as λ decreases the agreement between the computational curves and the analytically ones becomes clear. Furthermore the relation between the velocities of the two-phases verify the expected relation $u'_{\beta 0} = 0.1 u_{\alpha 0'}$.

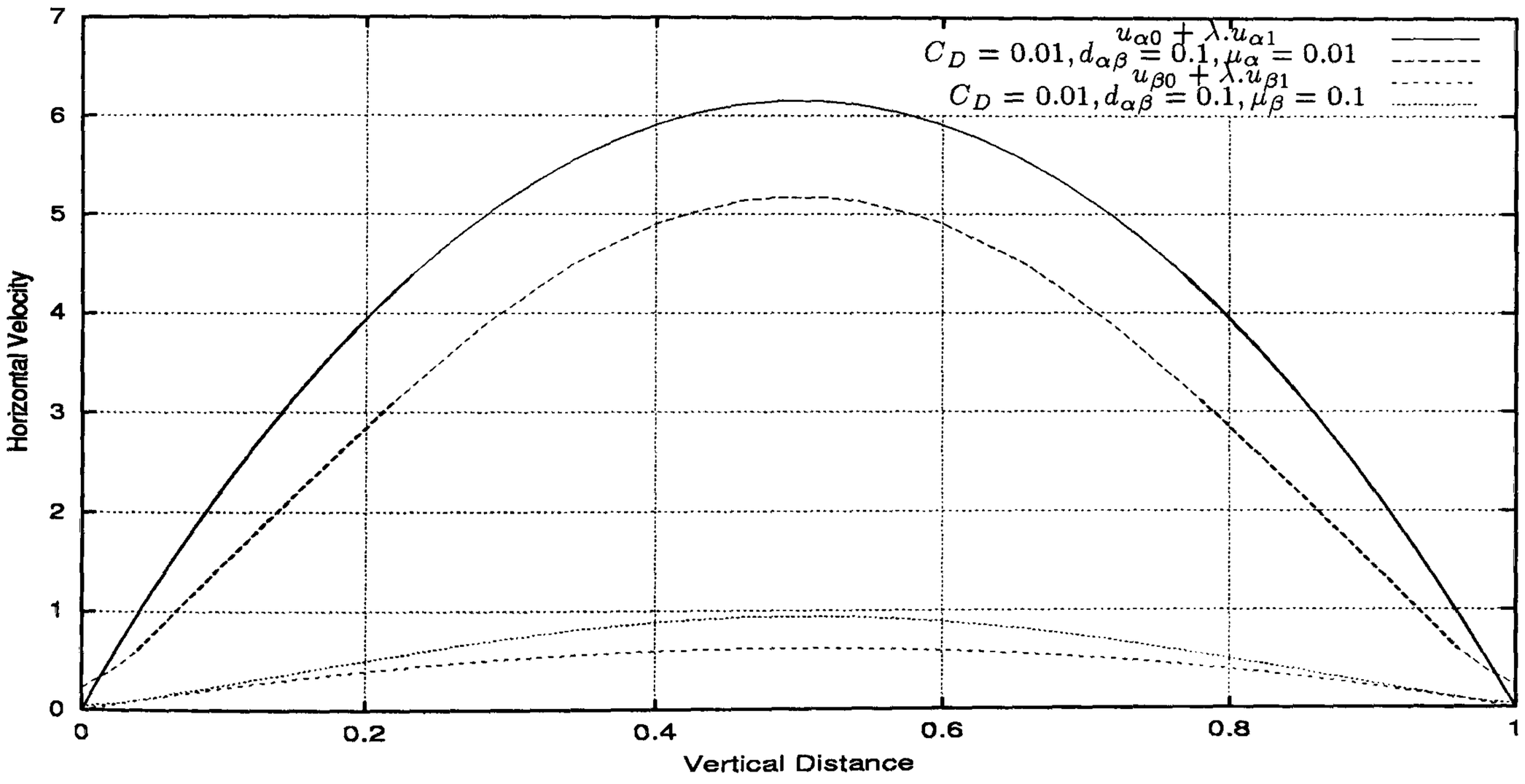


Figure 6.8: Two-phase channel flow - comparison between the analytical horizontal velocity and the computational one along the line $x = 3.0$ for $\lambda = 0.1$. It can be seen the same parabolic pattern of the analytical and computational results for both phases.

Figures 6.11, 6.12, 6.13 and 6.14 present the comparison between the volume fractions profiles computed by the `transient Pamg Multi-phase` and the corresponding analytical expressions derived in Section 6.3.1(Equations (6.70) and (6.71)) for the first test case outlined above and when $\lambda = 0.001$. To validate the discrepancies observed between the computed volume fractions and the analytical

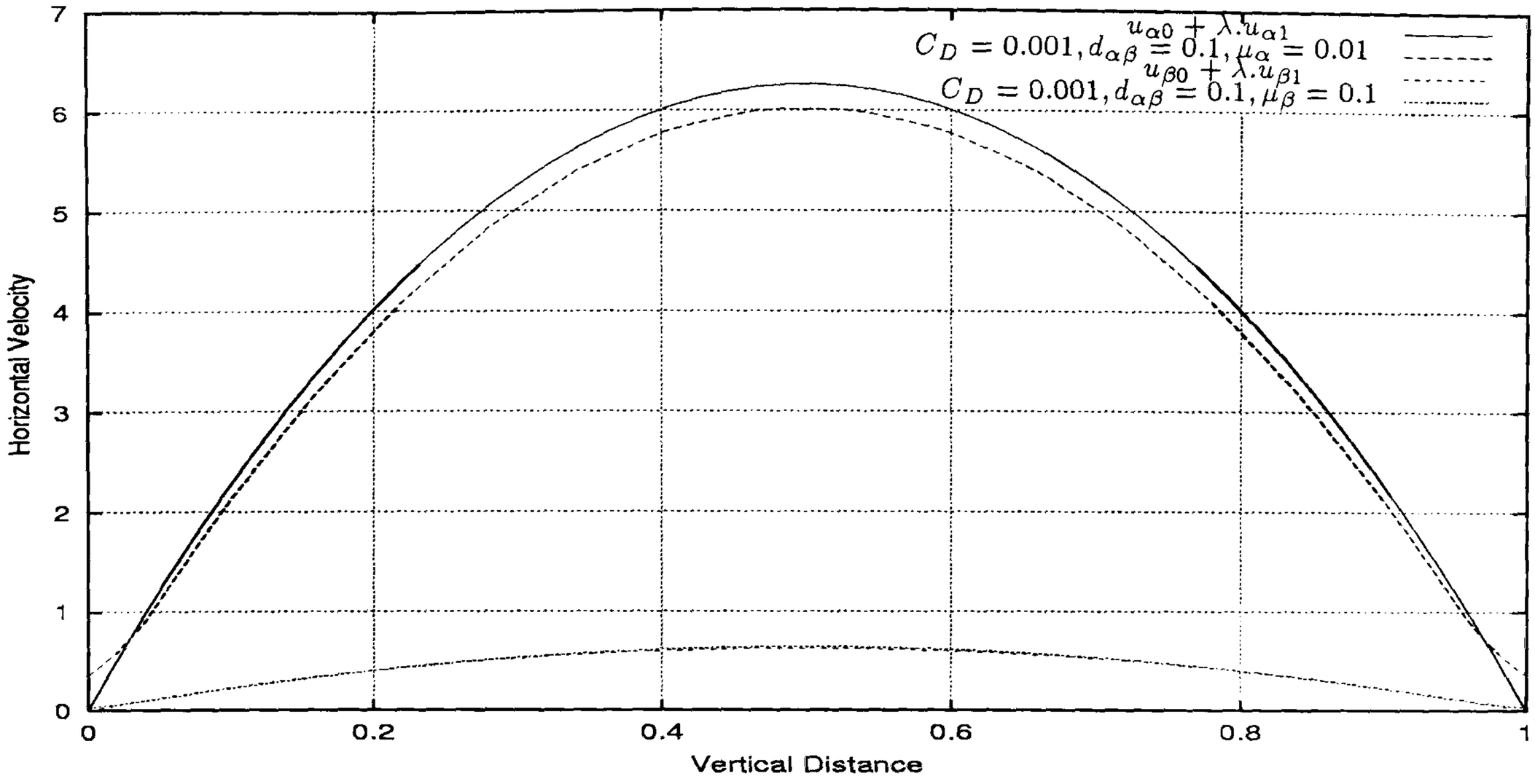


Figure 6.9: Two-phase channel flow - comparison between the analytical horizontal velocity and the computational one along the line $x = 3.0$ for $\lambda = 0.01$. Good agreement can be observed between the analytical and computational results for both phases.

ones in Figure 6.11 other experiments have been carry on with increasing length of the pipe. So Figure 6.12 represents the results for a pipe with the outlet at $x = 12.0$. Figure 6.13 the case for the outlet at $x = 24.0$ and finally Figure 6.14 the case for which the longer outlet has been considered. It can be verified that when the outlet is longer the agreement between the computed solutions and the analytically calculated ones is stronger. This may be due to two main reasons. One is the assumption of an expression for the volume fractions (Equation (6.63)) based on observations of the computed results, which are more accurate if the domain is longer. Another assumption related with the previous is that in a long pipe the variation of the velocity does not depend on the horizontal component. Hence the results observed suggest that if we choose a domain with a very long length the agreement between computational and analytical results will become better.

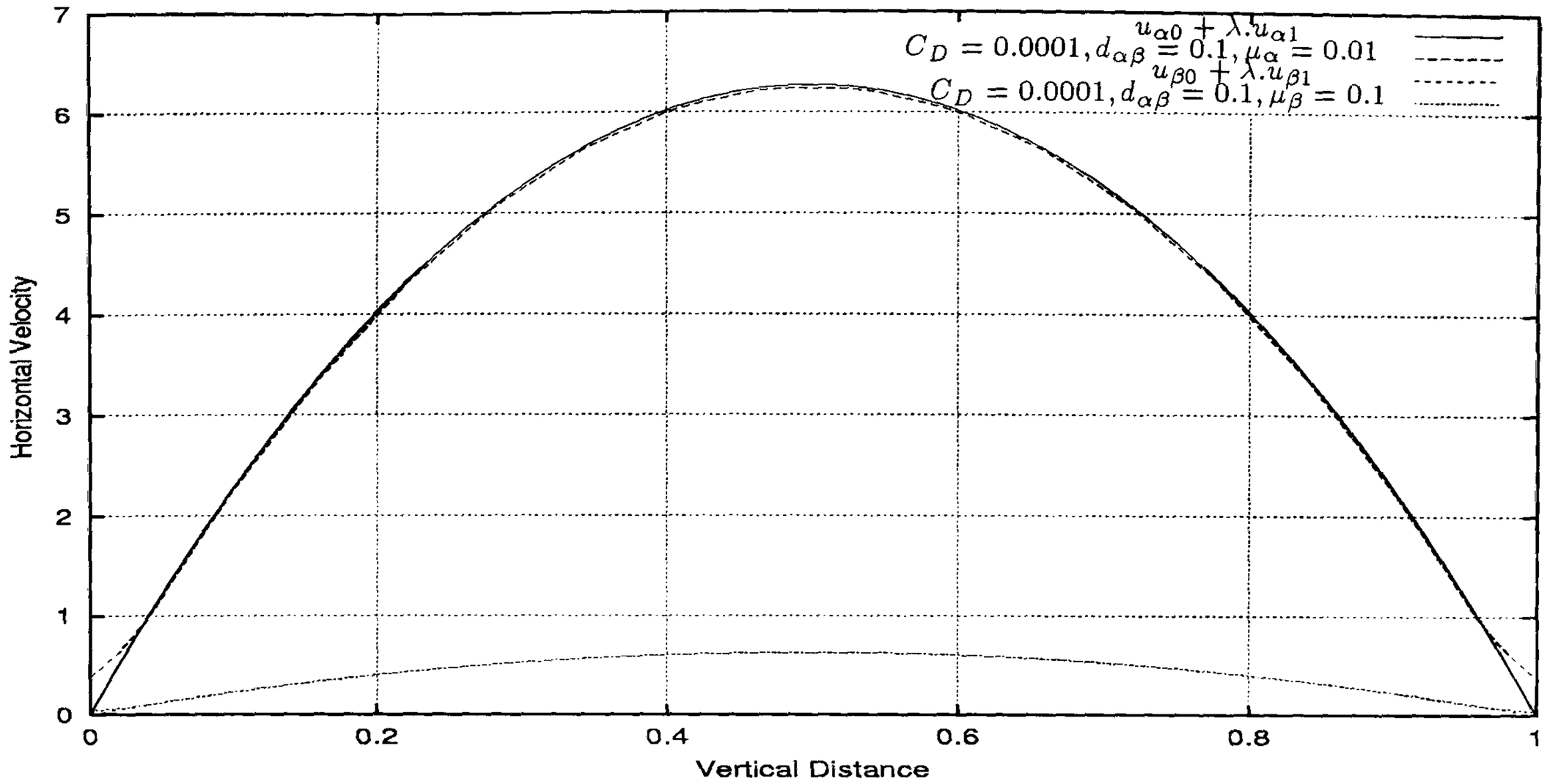


Figure 6.10: Two-phase channel flow - comparison between the analytical horizontal velocity and the computational one along the line $x = 3.0$ for $\lambda = 0.001$. It can be observed the excellent agreement between the analytical and computational results. The slight jumps in the computed results at the walls are due to interpolation errors.

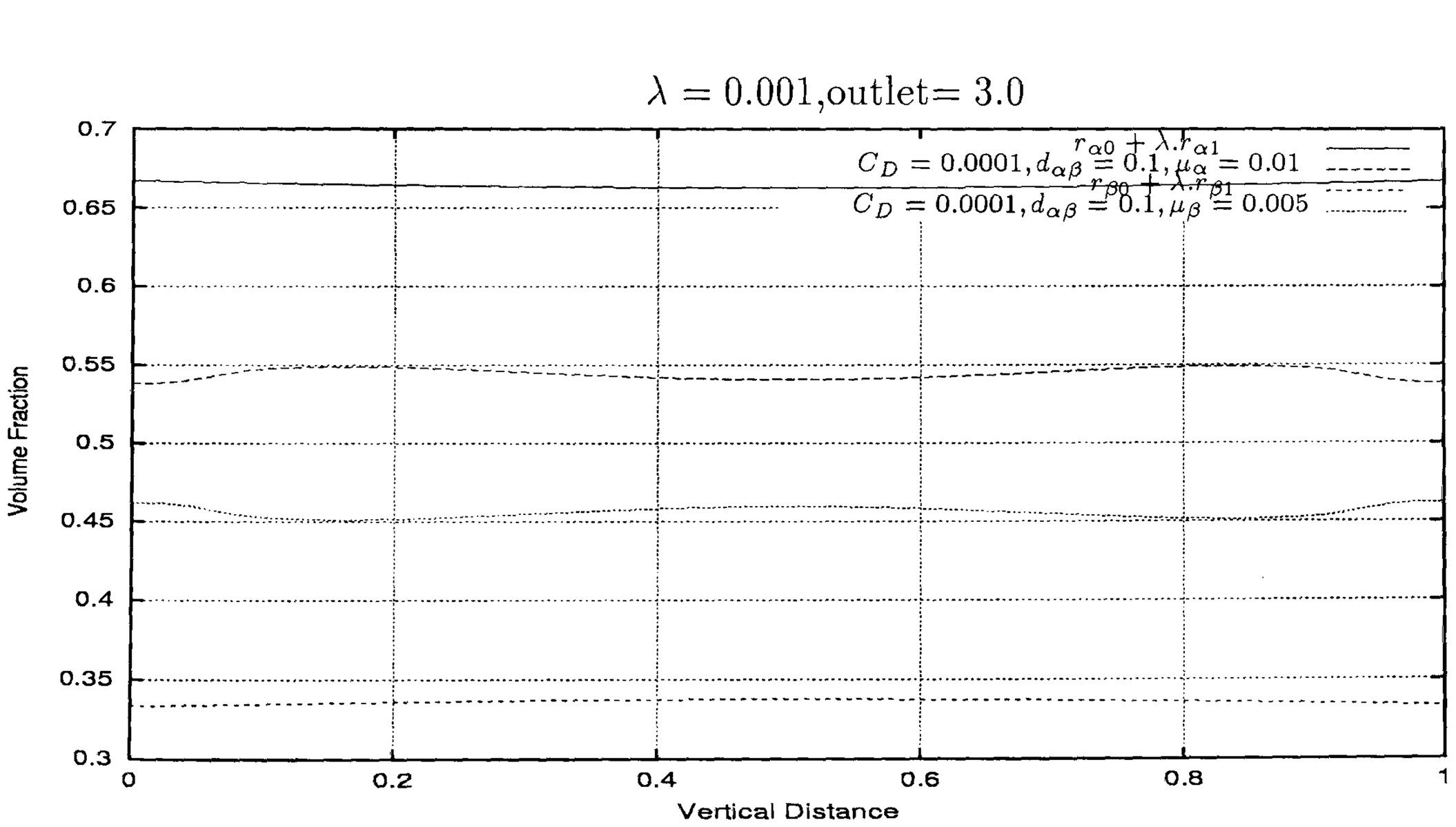


Figure 6.11: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 3.0$ for $\lambda = 0.001$. The analytical volume fractions exhibit a very small variation. This variation is bigger in the computational volume fractions.

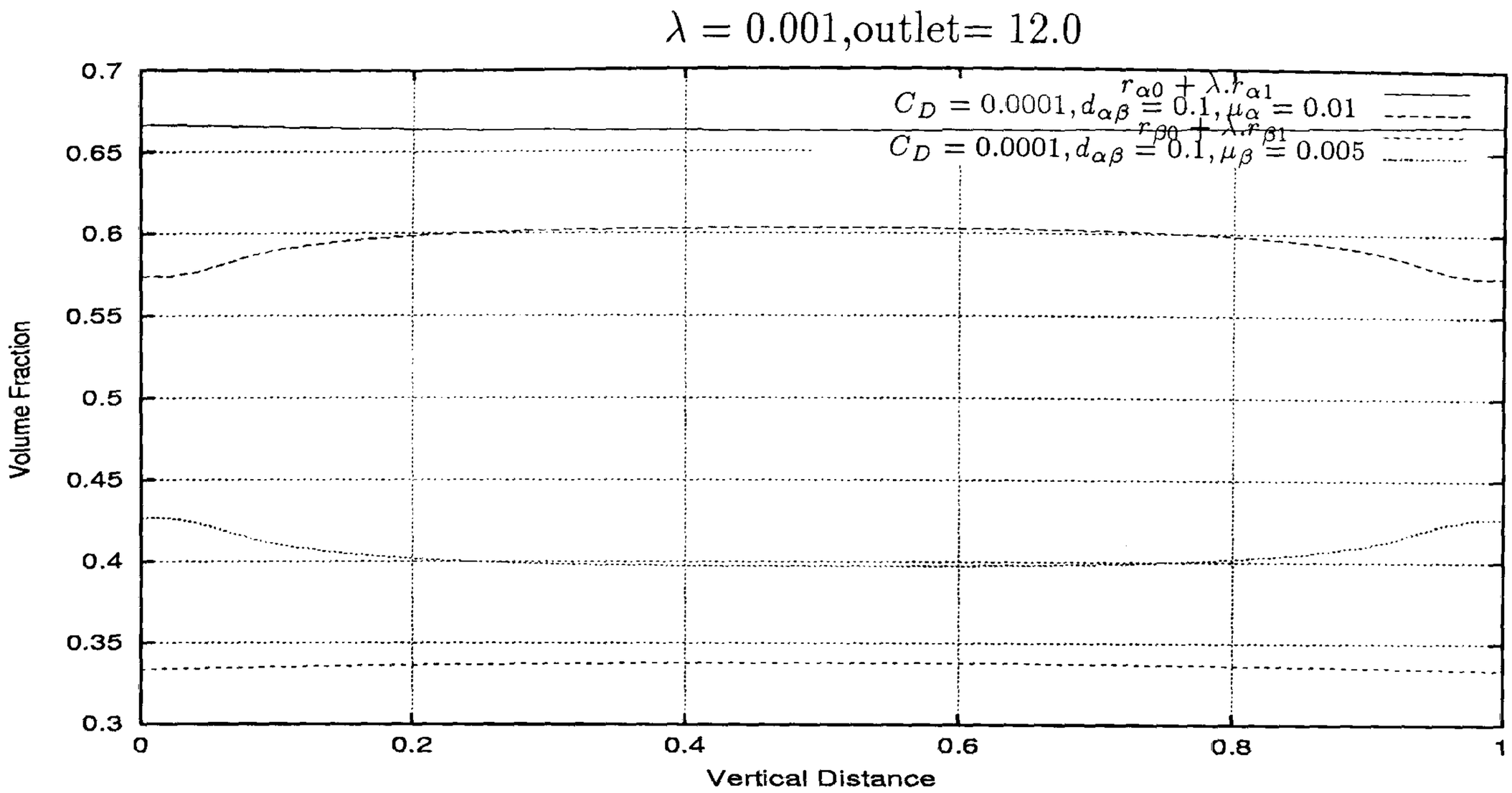


Figure 6.12: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 12.0$ for $\lambda = 0.001$. The disagreement between the analytical results and the computational ones is smaller then that one observed in the previous figure.

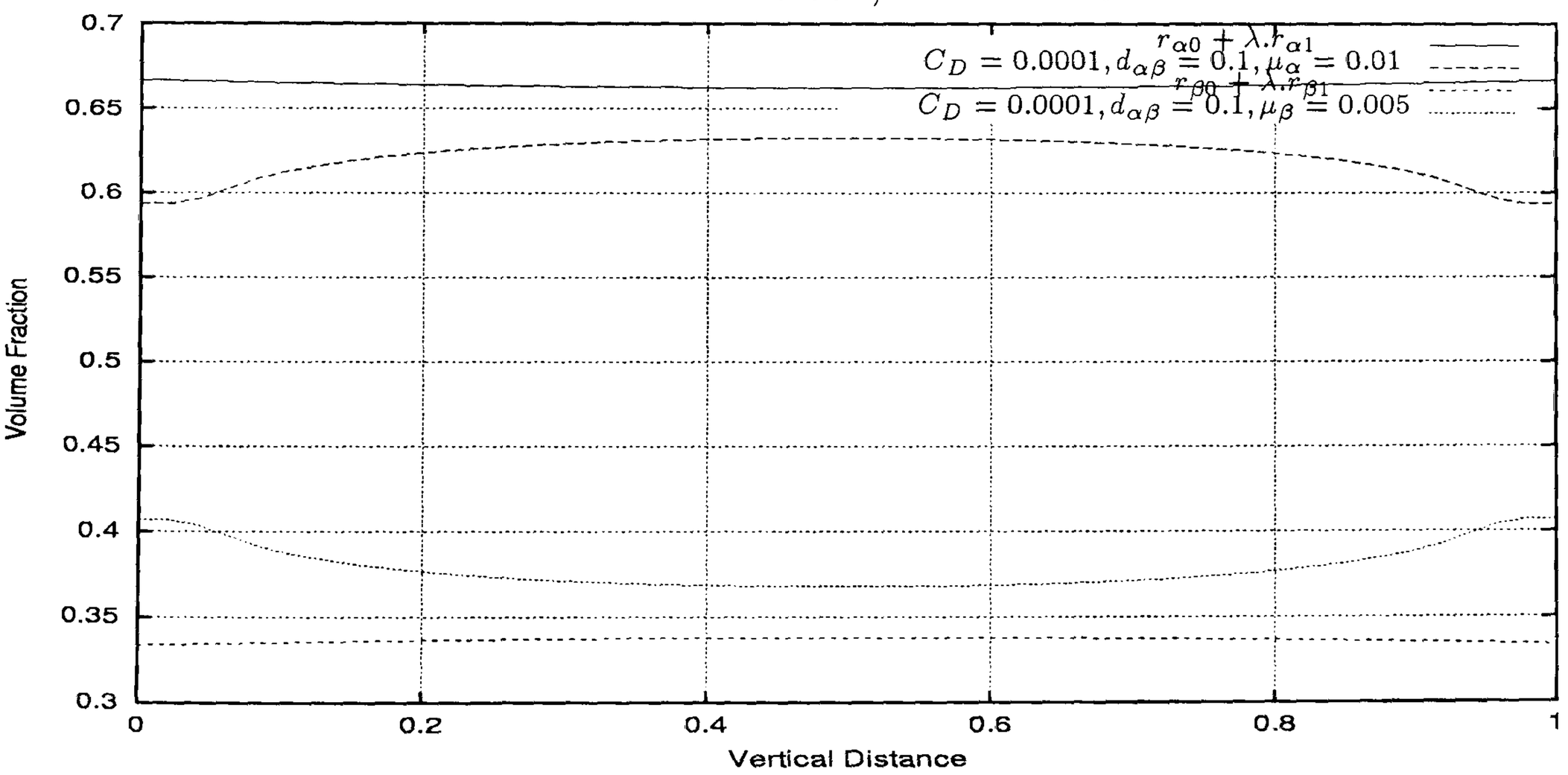


Figure 6.13: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 24.0$ for $\lambda = 0.001$. It is visible that with this longer geometry the analytical results and the computational ones becomes closer.

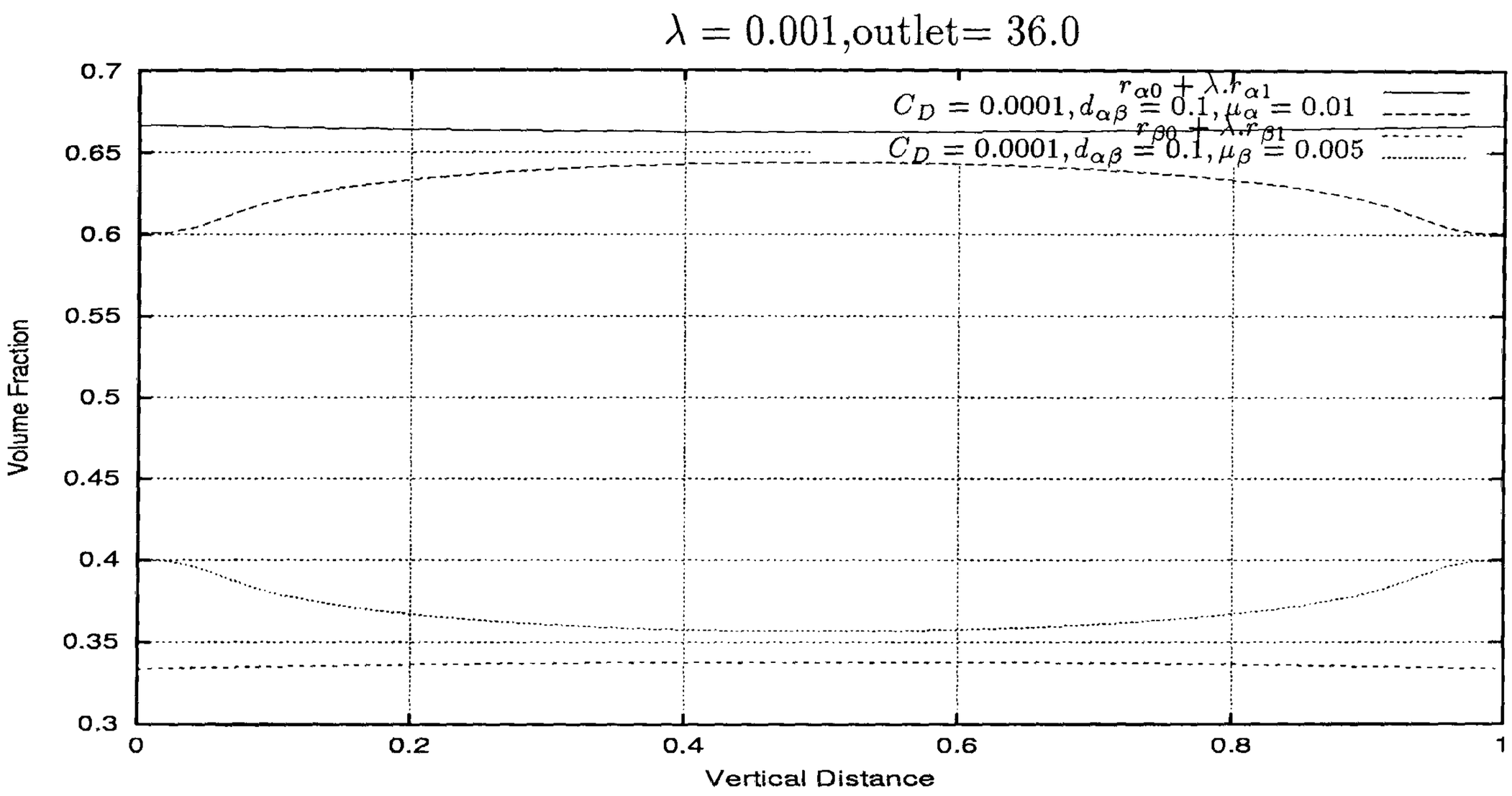


Figure 6.14: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 36.0$ for $\lambda = 0.001$. It is visible better agreement between the analytical results and the computational ones with this longer channel.

In Figures 6.15, 6.16, 6.17 and 6.18 which represent the same test cases outlined above but for the $\lambda = 0.0001$ case the same behaviour can be observed. In fact the agreement is slightly better due to the smaller λ used. The asymptotic formulas derived for the volume fractions give us a better approximation of the solution when the parameter λ is smaller.

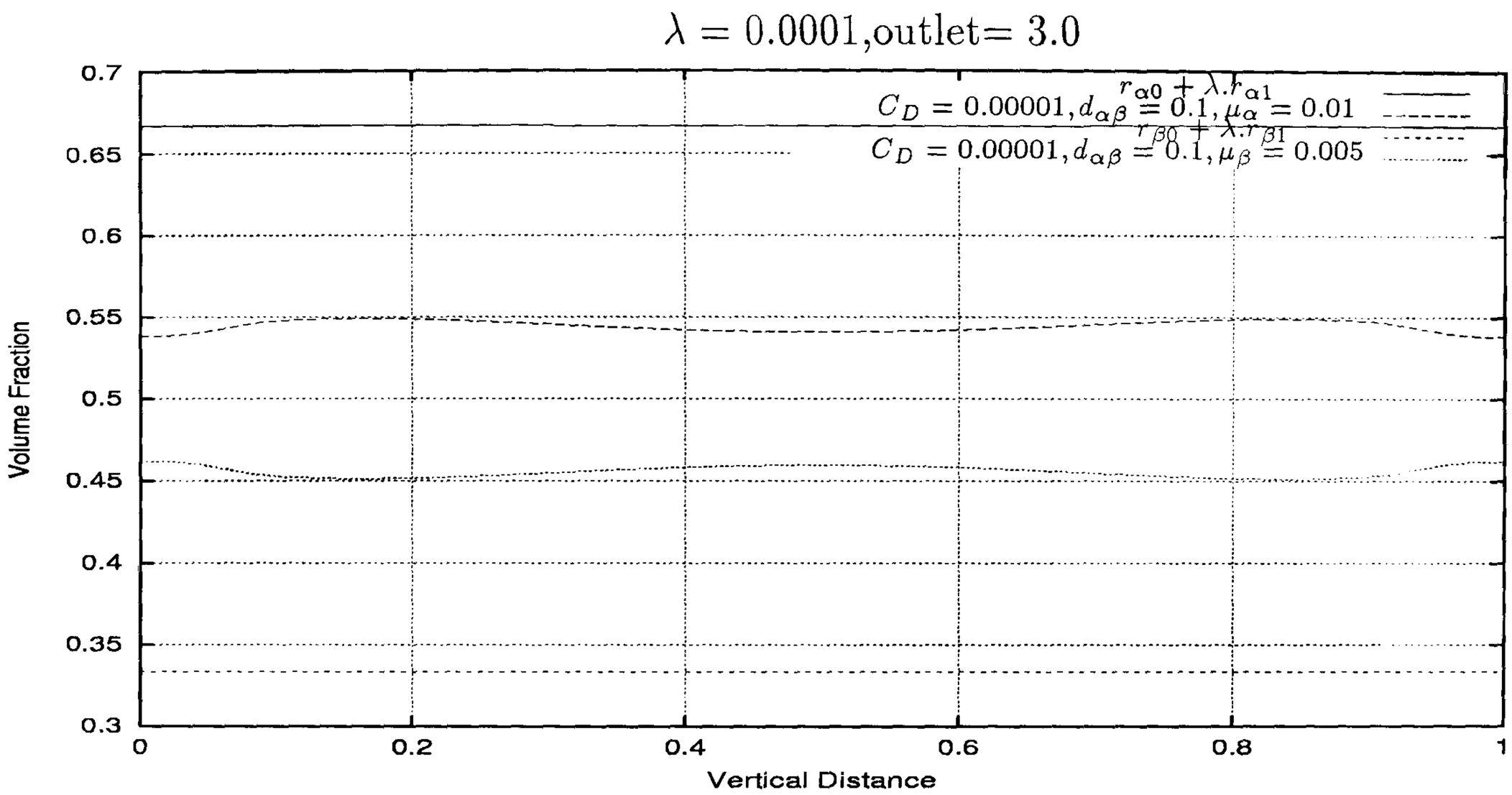


Figure 6.15: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 3.0$ for $\lambda = 0.0001$. The analytical volume fractions exhibit a very small variation while the computational volume fractions show a bigger variation.

Finally we present in Figures 6.19, 6.20, 6.21 and 6.22 the results for the second experiment outlined above when presenting the results for the velocities. Basically the differences to the first test case concerns the viscosity of the fluids chosen. As observed above when the domain becomes longer the analytical expressions of the volume fractions give solutions more close to the computed ones. The relevance of this test case relies on the bigger differences between the volume fractions of the two-phases. This is naturally due to the bigger differences in the Reynolds numbers of the two phases.

The main purpose of this section has been to validate the asymptotic formulas derived in Section 6.3.1. The methodology employed relies on the comparison of the analytical results with the results produced by `transient Pamg Multi-Phase` for a simple channel problem. Two test cases with fluids with different viscosities and same densities have been studied. The reason for choosing the same densities was to guarantee that there was no vertical mixing. Otherwise the volume fractions will depend on the vertical component. Different values for the inter-facial length and for the drag coefficient have been chosen. The values selected keep the parameter of the asymptotic expansion small so that the approximation of the velocities and of the volume fractions could be taken. It has been visible in the several figures

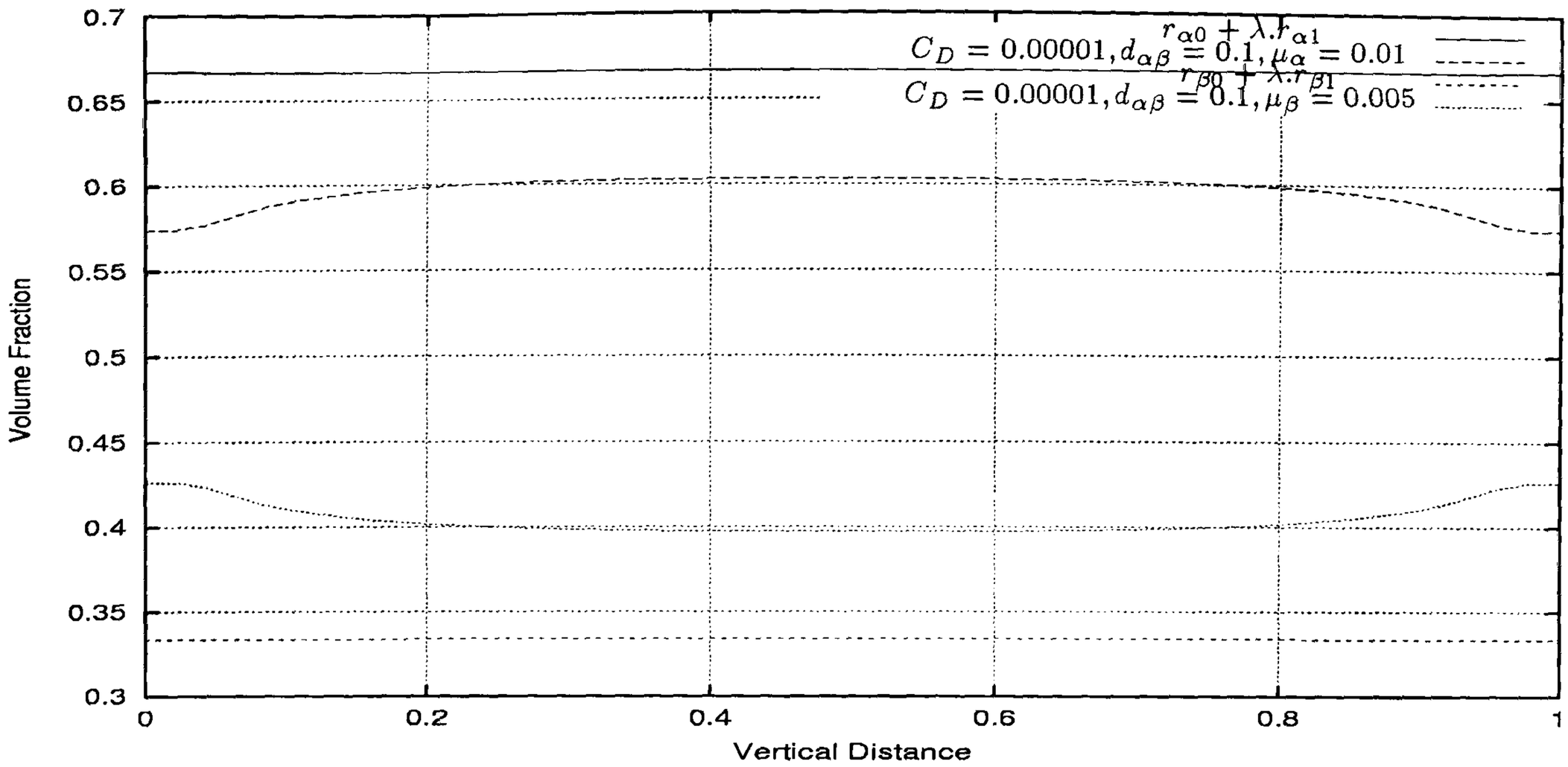


Figure 6.16: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 12.0$ for $\lambda = 0.0001$. The disagreement between the analytical results and the computational ones is smaller then that one showed in the preeceding figure.

presented that when the parameter of the expansion is smaller the agreement between the analytical results and the computational ones becomes better. This is particularly true with respect to the velocity profiles. The volume fractions profiles do not agree as well as the velocity profiles. Hence, further simulations have been produced with longer domains. It has been visible that as the domain becomes longer the agreement between the volume fractions becomes better. This suggests that the agreement will be good if we chose a longer domain.

So far the asymptotic formulas for the velocities and volume fractions have been validated with the computational results produced by a simple test problem. In the next section these formulas are going to be used in the interface treatment of our first coupling problem.

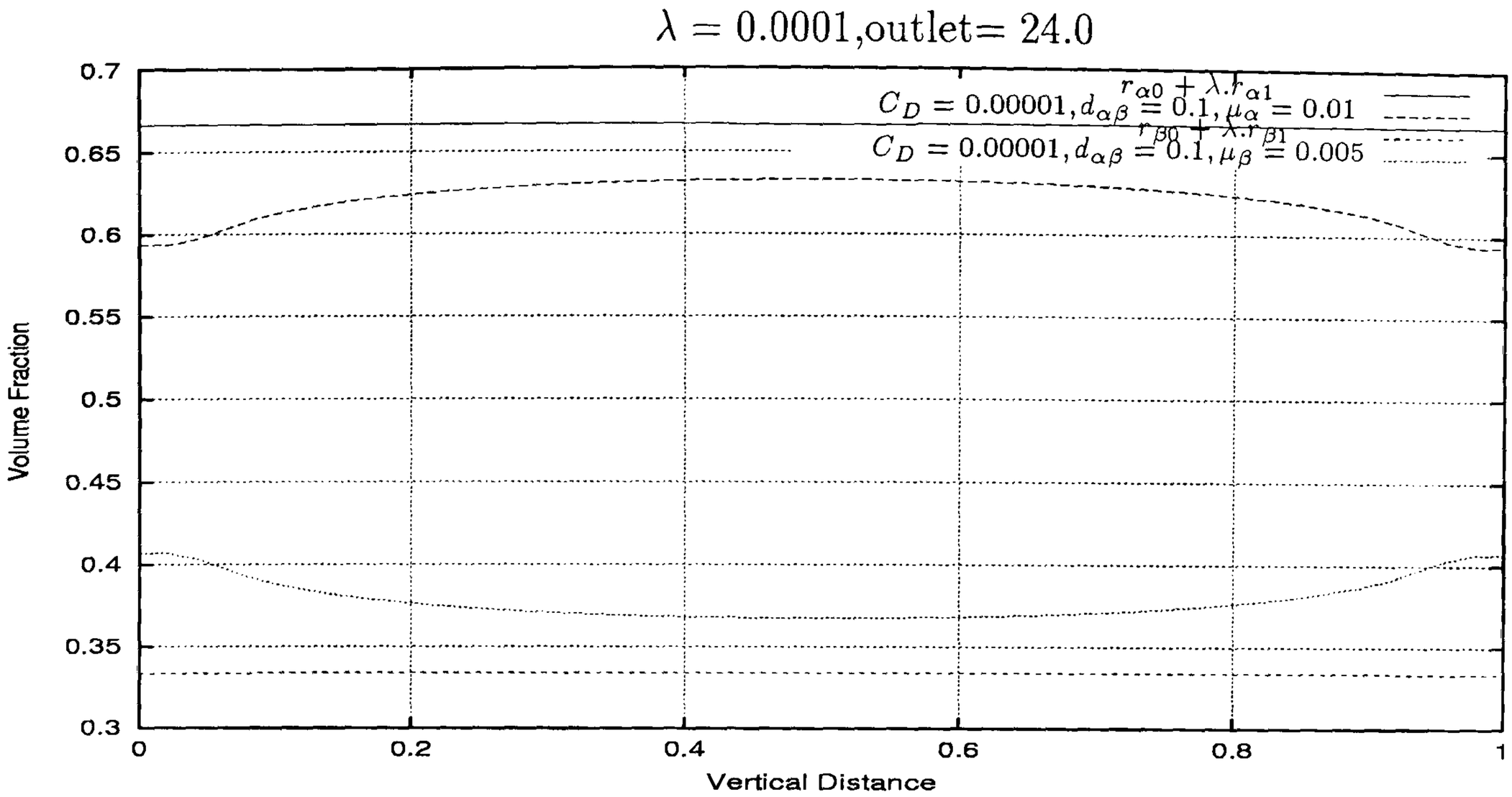


Figure 6.17: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 24.0$ for $\lambda = 0.0001$. It is visible that with this longer geometry the analytical results agree better with the computational ones.

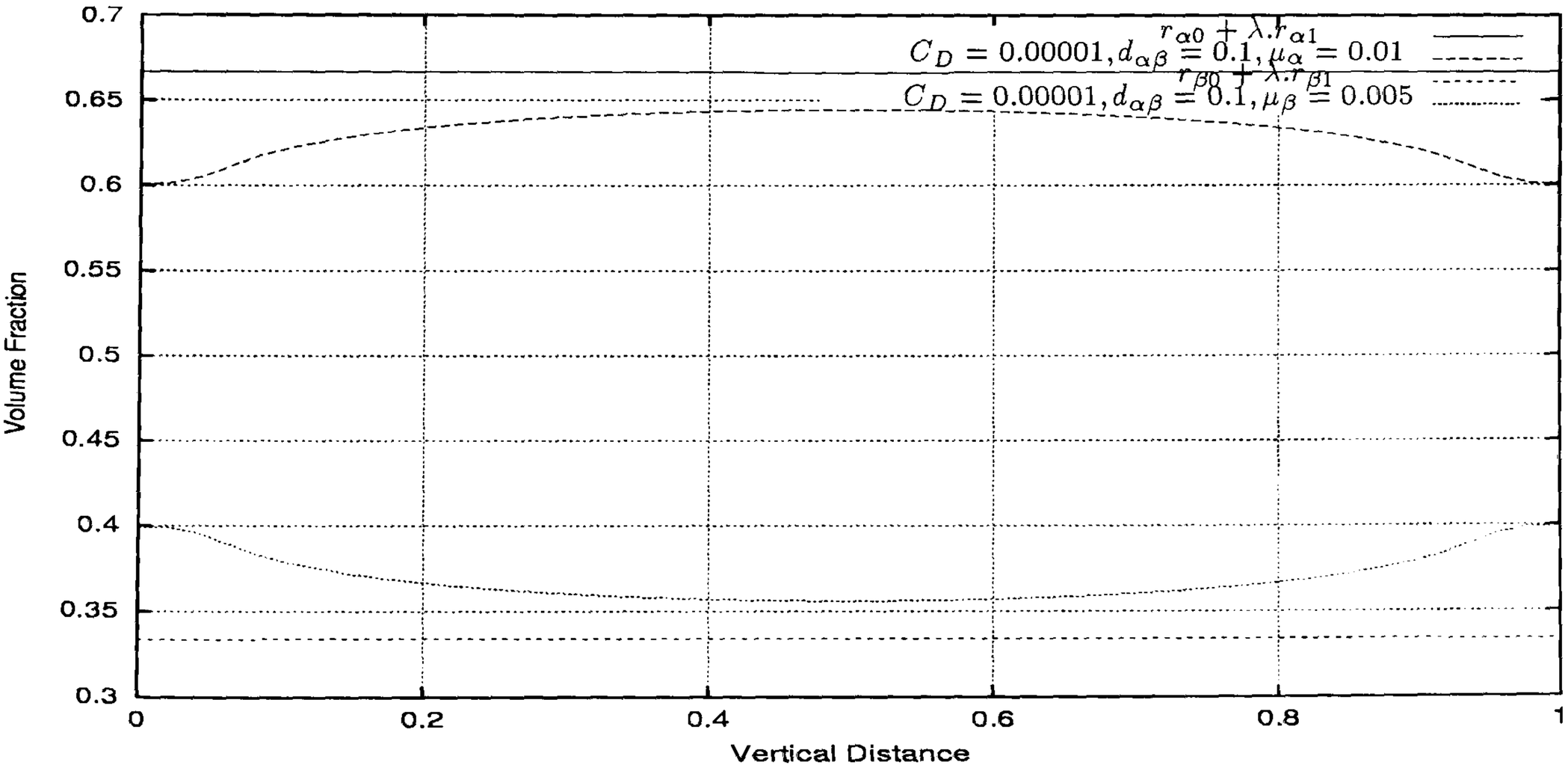


Figure 6.18: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 36.0$ for $\lambda = 0.0001$. It is visible better agreement between the analytical results and the computational ones with this longer channel

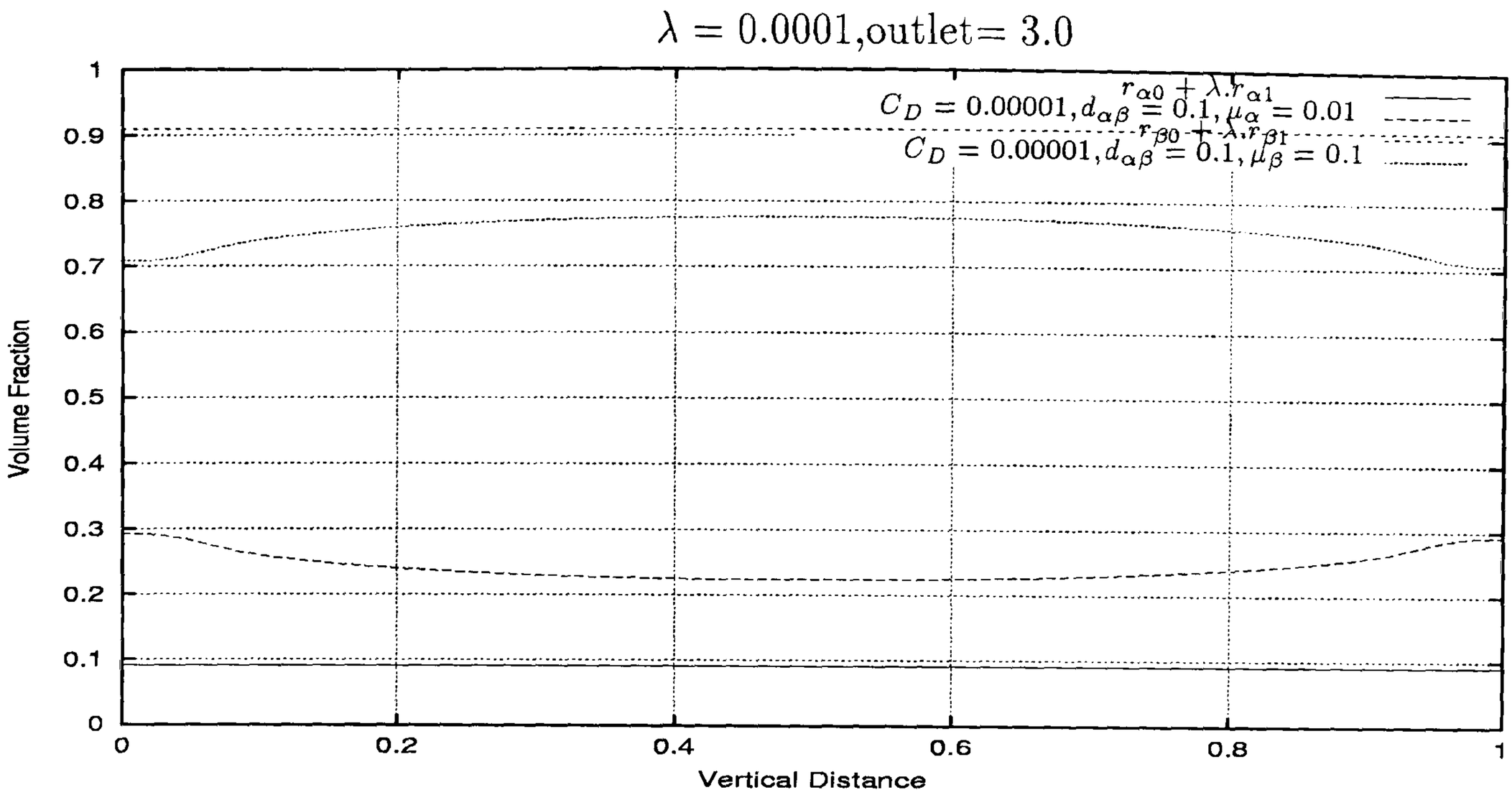


Figure 6.19: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 3.0$ for $\lambda = 0.0001$. The analytical volume fractions exhibit a very small variation while the computational volume fractions show a bigger variation

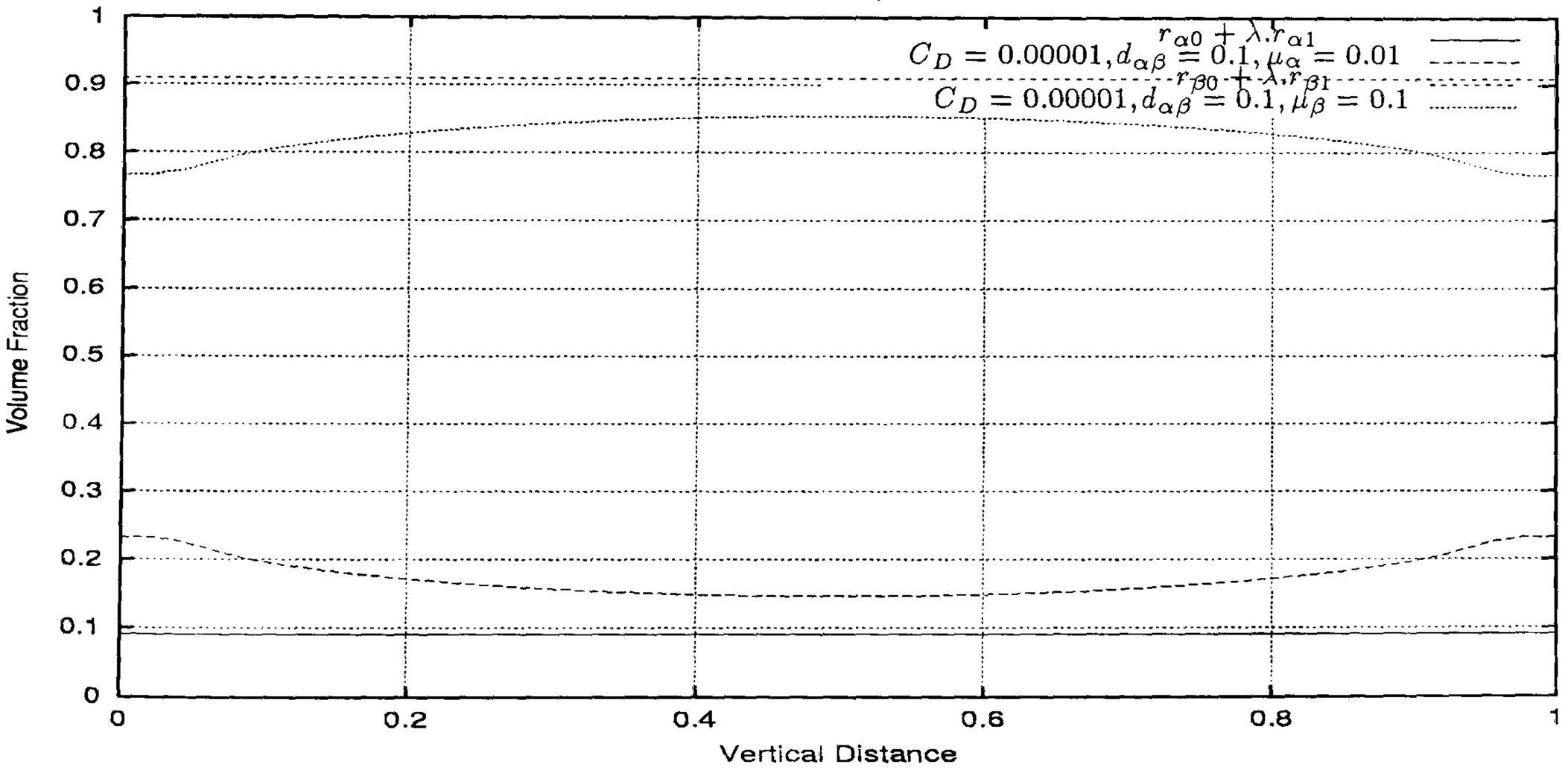


Figure 6.20: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 12.0$ for $\lambda = 0.0001$. The disagreement between the analytical results and the computational ones is smaller then that one showed in the preeceding figure.

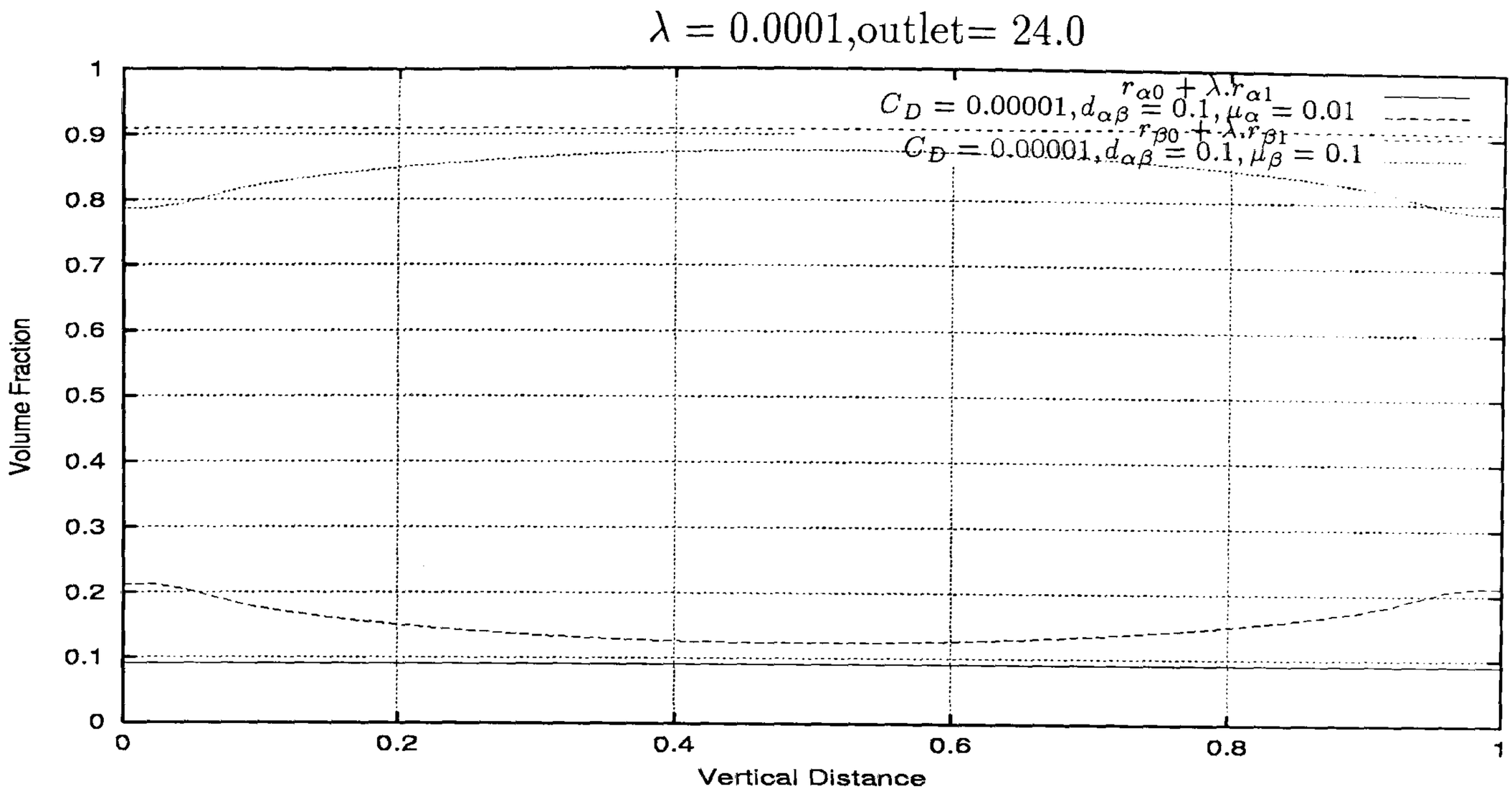


Figure 6.21: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 24.0$ for $\lambda = 0.0001$. It is visible that with this longer geometry the analytical results agree better with the computational ones.

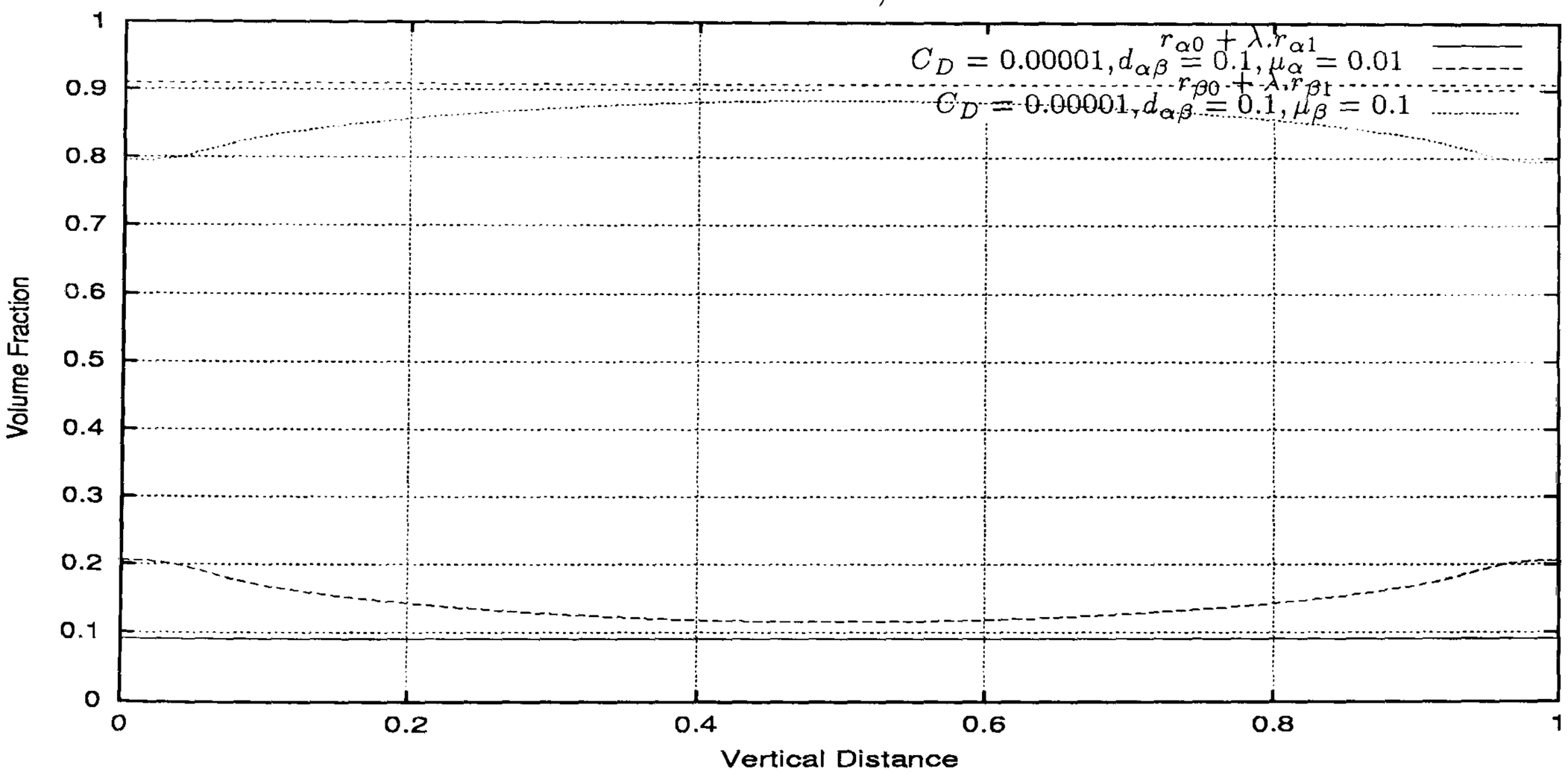


Figure 6.22: Two-phase channel flow - comparison between the analytical volume fraction and the computational one along the line $x = 36.0$ for $\lambda = 0.0001$. It is visible better agreement between the analytical results and the computational ones with this longer channel.

6.3.3 Coupling of the One-Dimensional Solver Emaps with the Two-Dimensional Solver transient Pamg Multi-phase

In this section we are going to couple the one-dimensional solver Emaps[16, 17, 78, 79] with the two-dimensional solver transient Pamg Multi-phase. Emaps has been designed to be directly applicable to a large number of multi-phase models with minimum code modification. It supports[79] Drift-flux, two-fluid and three-phase models. A complete description of Emaps can be found in references [16, 17]. The application of this solver to some particular problems can be found in references [78, 79]. The one-dimensional transient model Drift-flux has been the choice for this project. This coupling process originated our first version of the D-adaptive solver. The way in which the velocities and volume fractions are interchanged between the different dimensional regions is by an output/input file. This means that after computing the solutions in the one-dimensional region an output file is produced that is used as input file in the two-dimensional computation. This methodology is used exactly in the same way when changing from the two-dimensional region to the one-dimensional one. The validation of this D-adaptive solver is established by the comparison of the results produced by the simulation of a two-phase fluid flow through a contraction with the results produced by the transient Pamg Multi-phase code over the entire domain. The geometry of this test case is presented in Figure 6.23. Figure 6.24 shows the sections where the solution profiles are going to be presented.

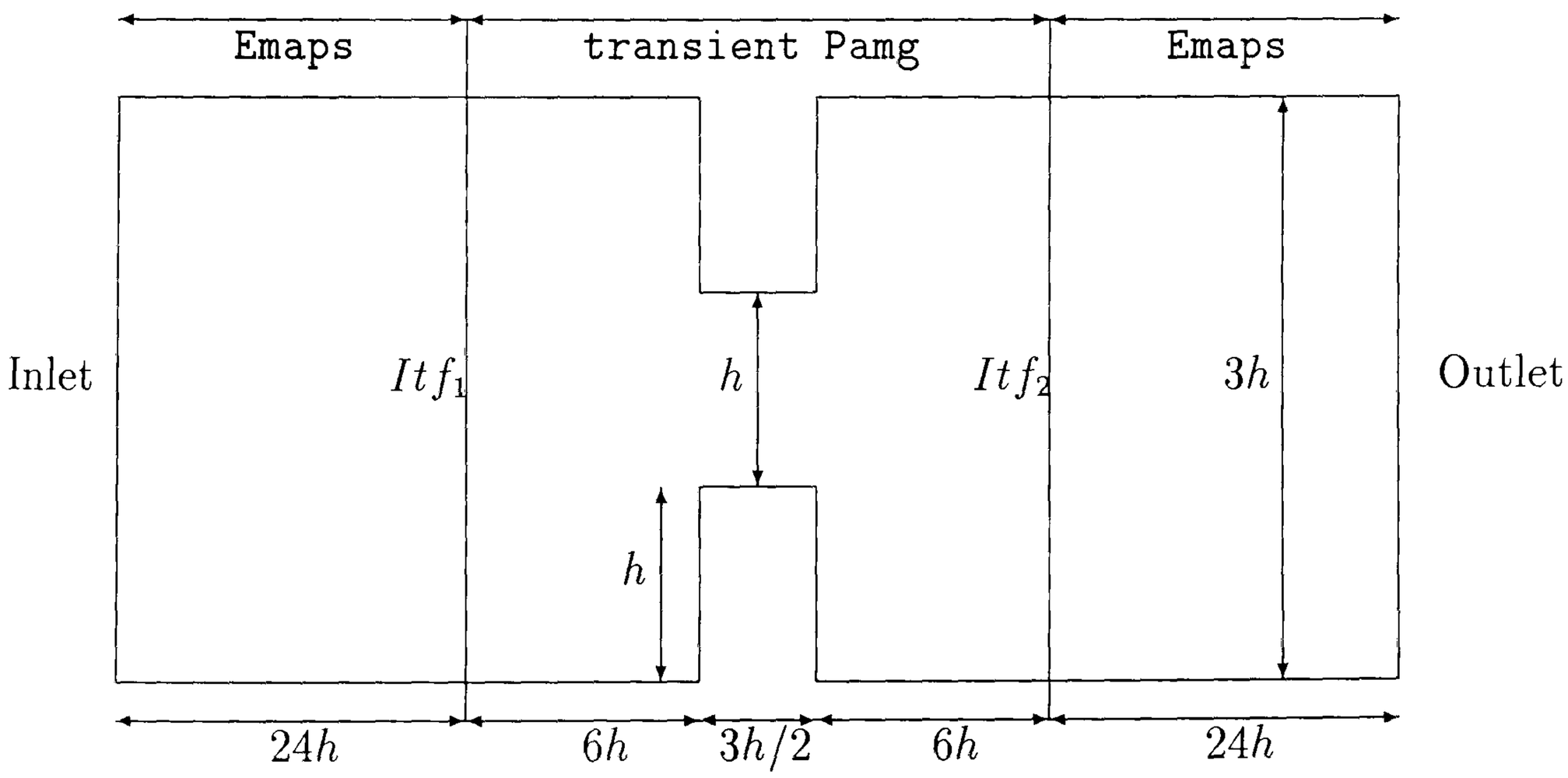


Figure 6.23: Geometrical representation of the domain of the Contraction flow problem

The physical properties chosen for this test case are:

The boundary conditions used are: At the inlet $(x = 0)$:

- Velocities $u(x) = 0.1$ for both phases

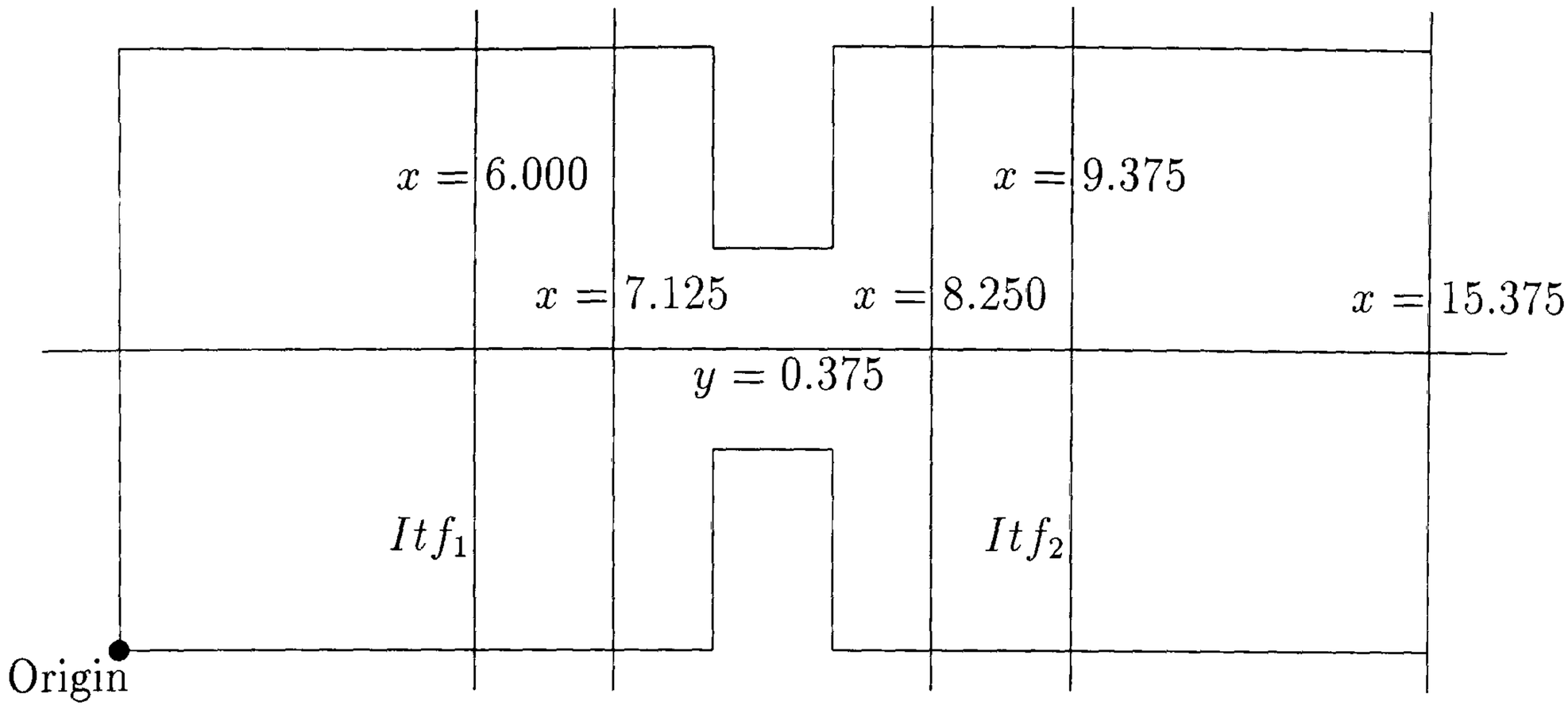


Figure 6.24: Contraction flow problem - main sections used to give the solution profiles

Two-Phase Flow	Reynolds Number	Viscosity	Density
Phase α	10	0.0075	1.0
Phase β	7.5	0.005	0.5

- Volume fractions $r(x) = 0.5$ for both phases

At the outlet: Neumann conditions.

The value chosen for the drag coefficient is $C_D = 0.0$. Hence the values chosen for $d_{\alpha\beta} = d_{\beta\alpha} = 0.1$ are irrelevant (see Equations 6.22 and 6.23).

The treatment of the interface Itf_1 (see Figure 6.23) is now outlined. Let $\overline{u(x)}$ and $\overline{r(x)}$ be the velocity and volume fractions for each phase computed by the one-dimensional code at point $x = 6.0$. Remembering that $3h$ is the height of the domain in the first one-dimensional region we obtain the following interpolation formulas:

$$\overline{u(x)} \times \overline{r(x)} = \frac{1}{3h} \int_0^{3h} urdy$$

where u and r represent the velocity and volume fractions of each phase in the two-dimensional solver `transient Pamg Multi-phase`. Hence to get a parabolic profile in the line $x = 6.0$ it has to be verified the relation

$$\frac{1}{3h} \int_0^{3h} urdy = \frac{1}{3h} \int_0^{3h} coef(3hy - y^2)rdy$$

The coefficient $coef$ must be set to $\frac{2\overline{u(x)} \times \overline{r(x)}}{3rh^2}$.

Another possible approach which we have used in this experiment is to use the

asymptotic formulas derived in Section 6.3.1 namely the equations

$$u_{\alpha 0'} = \frac{1}{2\mu_\alpha}\frac{\partial p^{'}}{\partial x^{'}} y^{'}(y^{'} - 3h)$$

$$u_{\beta 0'} = \frac{1}{2\mu_\beta}\frac{\partial p^{'}}{\partial x^{'}} y^{'}(y^{'} - 3h) \ .$$

where $3h$ is the height of the domain.

Hence we have to compute an approximation for $\frac{\partial p^{'}}{\partial x^{'}}$ which can be done through the formula

$$\frac{\partial p^{'}}{\partial x^{'}} = \frac{-12\mu_\alpha Q^{'}_{\alpha} r_{\alpha 0}}{3h^3}$$

where the flux $Q^{'}_{\alpha}$ is approximated by the expression $\overline{u(x)} \times \overline{r(x)}$ which gives the flux computed for each phase by the one-dimensional solver.

In interface Itf_2 the interpolation formulas used are:

$$\overline{u(x)} = \frac{\sum_{i=1}^{n} u_i \Delta x_i}{\sum_{i=1}^{n} \Delta x_i}$$

and

$$\overline{r(x)} = \frac{\sum_{i=1}^{n} r_i \Delta x_i}{\sum_{i=1}^{n} \Delta x_i}$$

where n is the number of cells and Δx_i is the lengh size of each cell. Since we use a uniform grid this expressions can be simplified by

$$\overline{u(x)} = \frac{\sum_{i=1}^{n} u_i}{n}$$

and

$$\overline{r(x)} = \frac{\sum_{i=1}^{n} r_i}{n}$$

In order to guarantee that in the second one-dimensional domain the relation between the velocities of the two phases are kept the `D-adaptive solver` has to compute the new constants to use in the hydrodynamic closure law in the Drift-Flux model (see Equation 2.17)). These constants C_0 and C_1 are updated by the following relation[27]

$$u_\beta = C_0(r_\beta u_\beta + r_\alpha u_\alpha) + C_1$$

where u_α, u_β, r_α and r_β are the one-dimensional velocities and volume fractions for phases α and β.

In Figure 6.25 we present the profiles of the horizontal velocities for *Phase* α and *Phase* β along the line $y = 0.375$ produced by the `D-adaptive solver` and the `transient Pamg Multi-phase`. We can see some good agreement for both the profiles until the middle of the domain. However afterwards the same behaviour is not observed. Particularly evident is the fact that the curve of each phase produced by one of the solvers is more close to the curve of the other phase produced by the other solver. This seems to suggest that there is a variation in the way that each

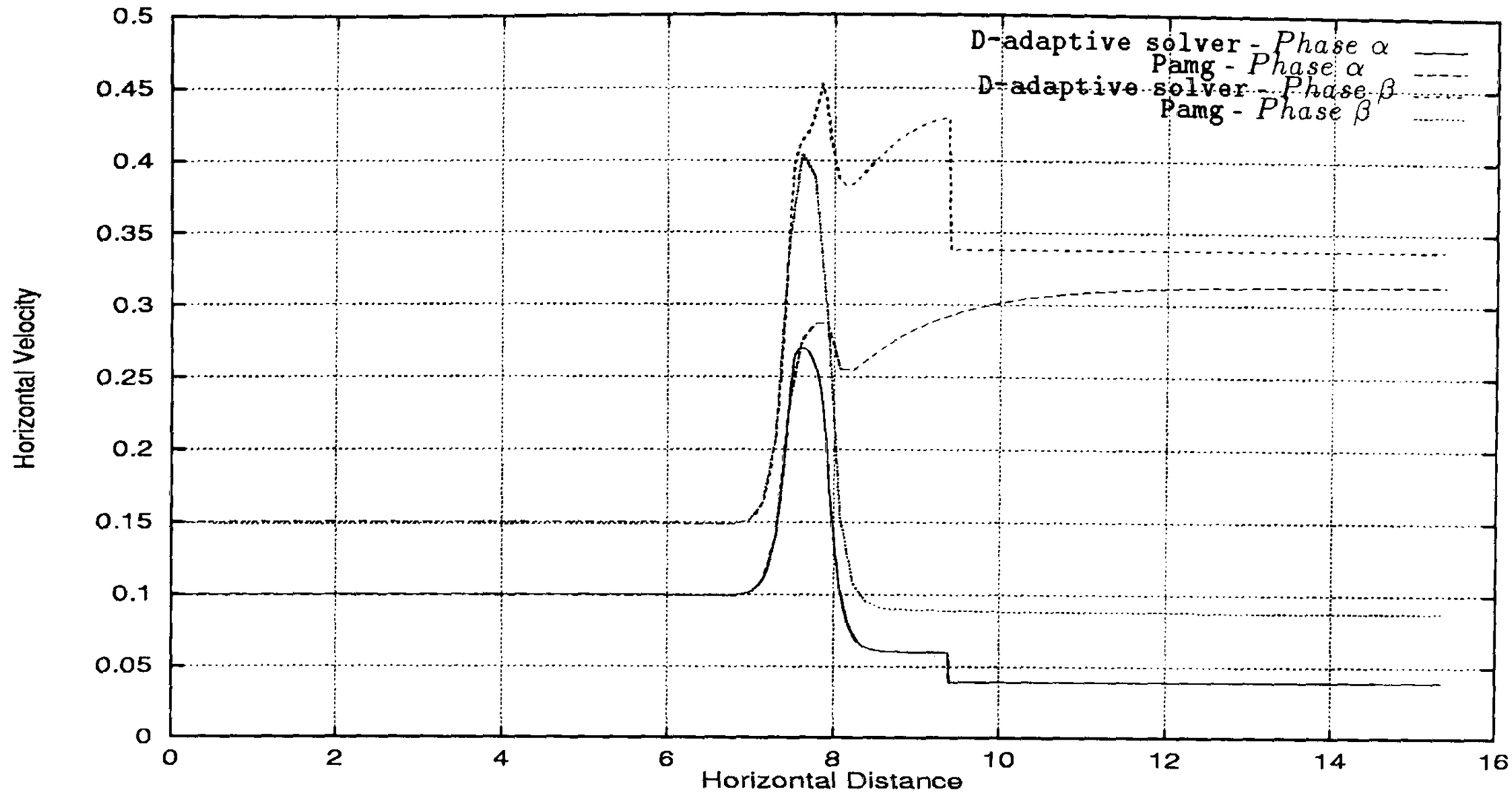


Figure 6.25: Two-phase flow through a Contraction (Case I) - comparison between the horizontal velocity profile along the line $y = 0.375$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen good agreement for both the profiles until the middle of domain. Afterwards the curves do not match exactly. It is visible the jump in the profiles of the `D-adaptive solver` at the second interface line

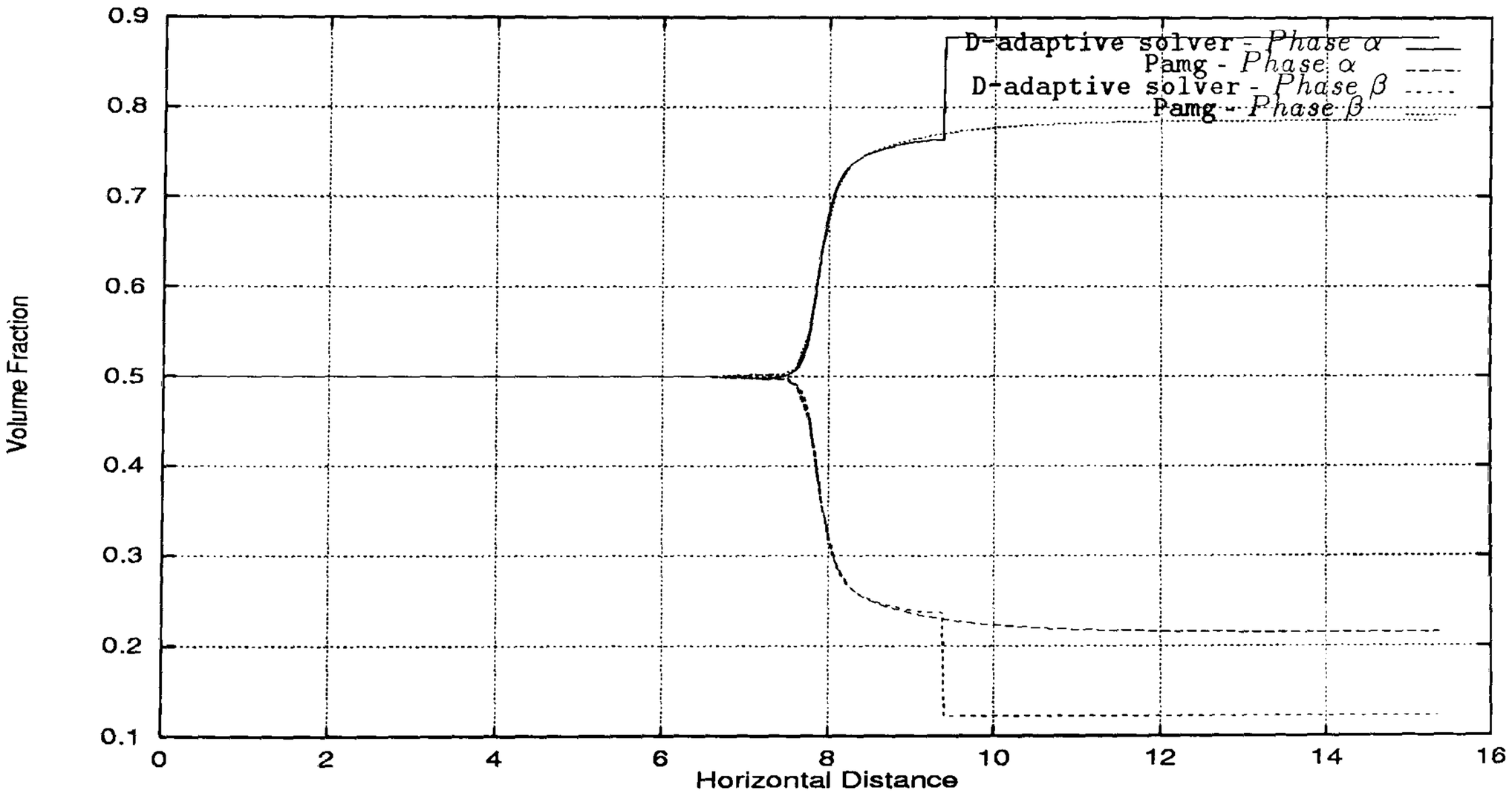


Figure 6.26: Two-phase flow through a Contraction (Case I) - comparison between the volume fraction profile along the line $y = 0.375$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It is visible the jump in the profiles of the `D-adaptive solver` at the second interface line.

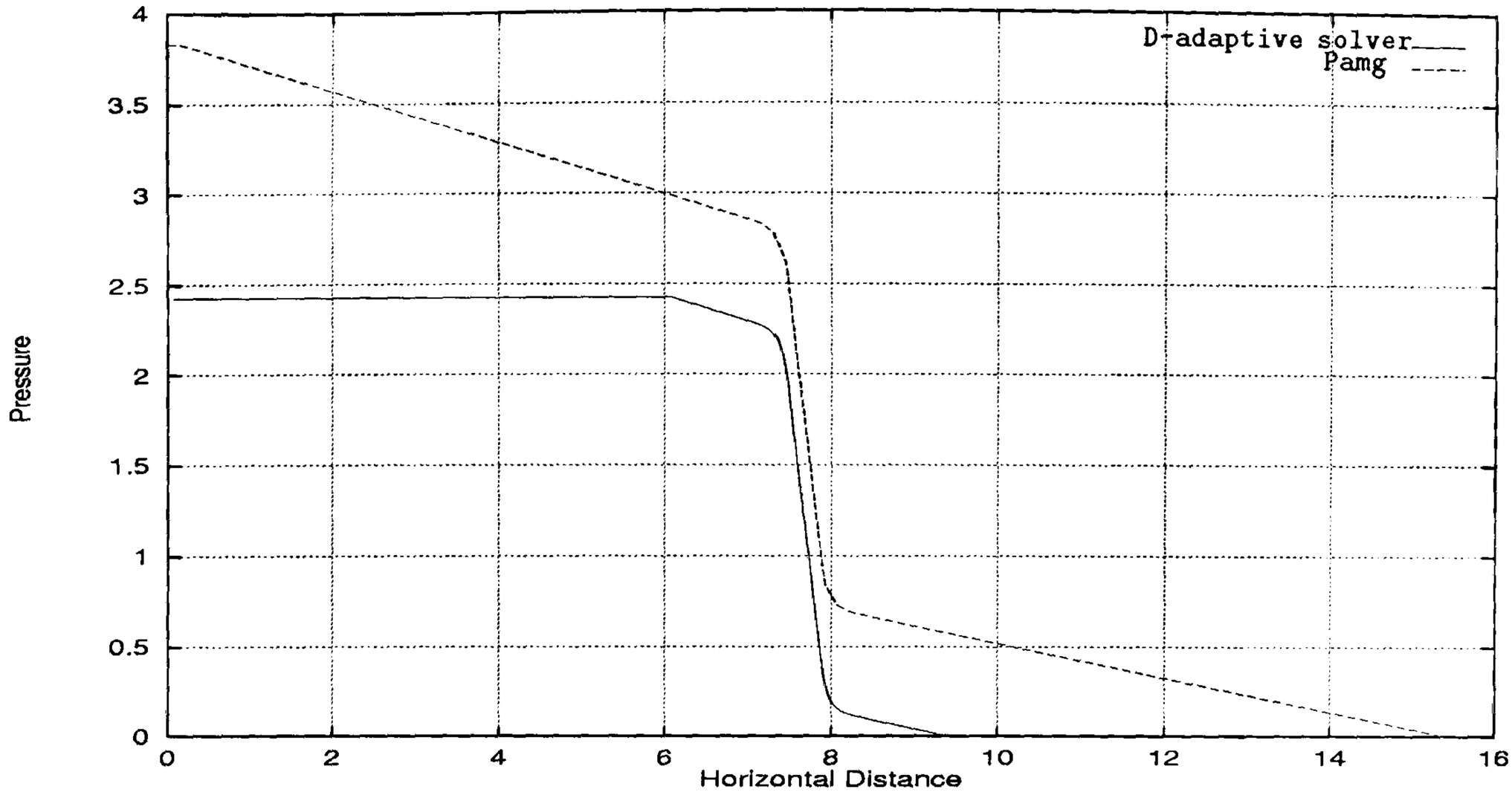


Figure 6.27: Two-phase flow through a Contraction (Case I) - comparison between the pressure drop profile along the line $y = 0.375$ for `D-adaptive solver` and `transient Pamg Multi-phase`. Good agreement can be observed in the contraction region between the two curves. In the one-dimensional regions the curves do not match exactly. Some pressure drop is observed in the `transient Pamg Multi-phase` code. No pressure drop is observed in the `D-adaptive solver` at those one-dimensional regions.

phase progresses after reaching the critical point corresponding to the maximum value.

Figure 6.26 show the profile of the volume fractions for *Phase* α and *Phase* β along the line $y = 0.375$ produced by the `D-adaptive solver` and the `transient Pamg Multi-phase`. Like in the velocities profiles the volume fractions change in a different way after the contraction region. *Phase* α on the `D-adaptive solver` goes up while on `transient Pamg Multi-phase` goes down. The opposite behaviour is observed on *Phase* β.

Figure 6.27 represents the pressure drop along the line $y = 0.375$ measure by the `D-adaptive solver` and the `transient Pamg Multi-phase`. Although in the contraction region it is possible, after shifting, to get a good agreement in the curves it is clear that in the first and second one-dimensional region the curves do not match. However, `transient Pamg Multi-phase` is a solver designed for Cartesian coordinates. `Emaps` is a code designed to work with pipes and consequently it is not prepared to work with cartesian geometries. The differences in the mass fluxes computed by both solvers originate some important difference on the results.

In Figure 6.28 we present the results for the horizontal velocity along the line $x = 6.0$, which has been defined as the first interface between the one-dimensional region with the two-dimensional one. It is visible the parabolic shape of the horizontal

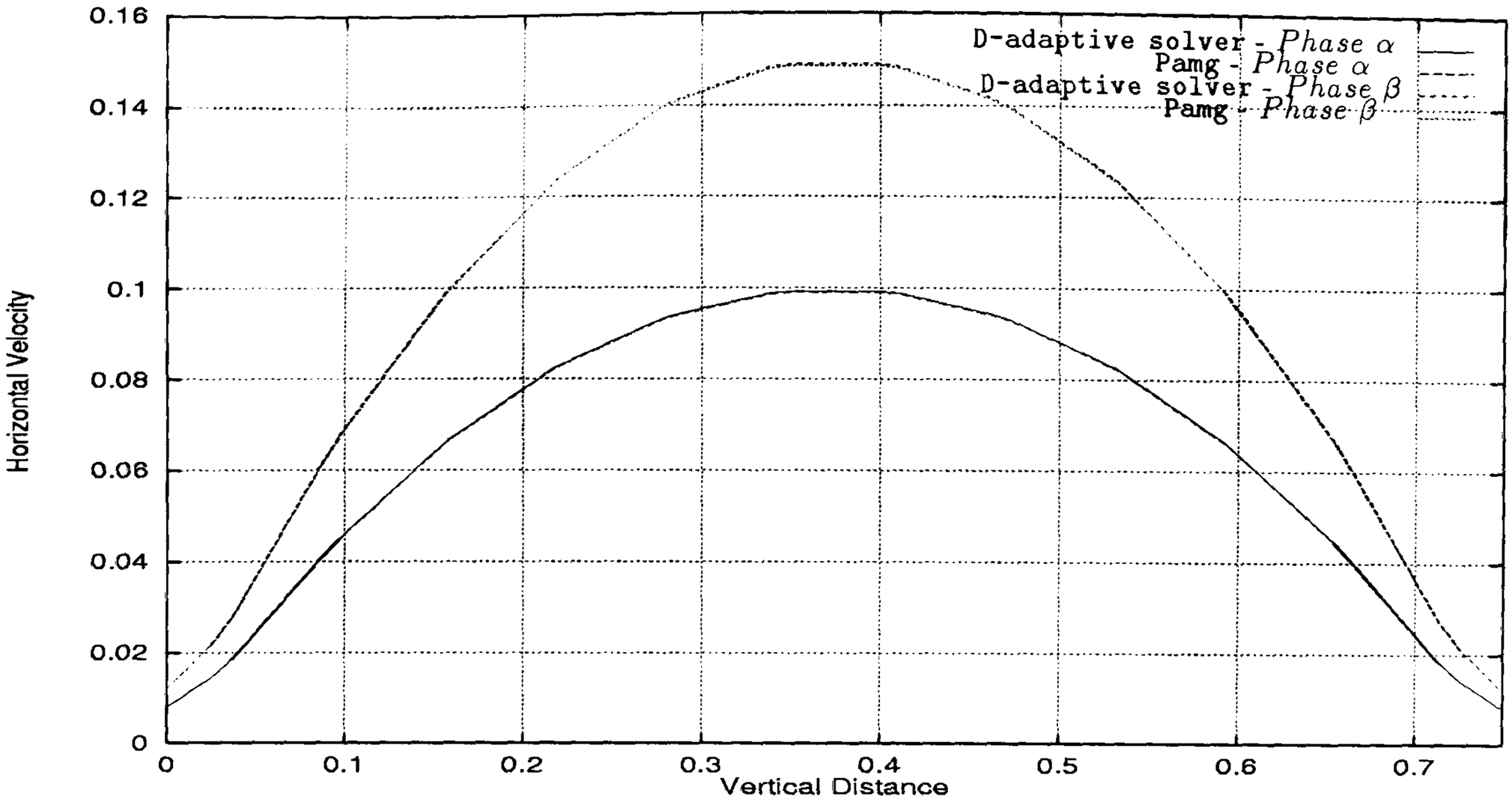


Figure 6.28: Two-phase flow through a Contraction (Case I) - comparison between the horizontal velocity profiles along the line $x = 6.000$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the parabolic profile of the horizontal velocity and the excellent agreement between the two solvers.

velocity. The volume fractions along this line are very close to 0.5.

The next set of figures gives the profiles for the horizontal velocities, vertical velocities and volume fractions along the lines $x = 7.125$ and $x = 8.250$. These two lines have been chosen because the line $x = 7.125$ is a short distance before the contraction and the line $x = 8.250$ after the contraction. Hence it is expected that we can see bigger variations on the velocities and volume fractions. Figures 6.29, 6.30 and 6.31 present the results along the line $x = 7.125$. We can see an excellent agreement between the horizontal velocities, vertical velocities and volume fractions between the `D-adaptive solver` and the `transient Pamg Multiphase`. In Figures 6.32, 6.33 and 6.34 the results along the lines $x = 8.250$ are shown. The horizontal velocities do not agree perfectly. However, they exhibit the same parabolic pattern. Furthermore the region that we are plotting is a very difficult one since it is exactly after the contraction region. So the variation in the horizontal velocities are expected to be bigger and consequently the agreement between the results produced by two different solvers is not expected to be great. The vertical velocities profiles along the line $x = 8.250$ are presented in Figure 6.33. The reasons outlined above in the horizontal velocities discussion justify some differences observed in the curves representing the results of the `D-adaptive solver` and the `transient Pamg Multi-phase`. Finally in Figure 6.33 the agreement between the volume fractions profiles produced by both codes is excellent.

Figures 6.35, 6.36 and 6.37 present the results for the horizontal velocities,

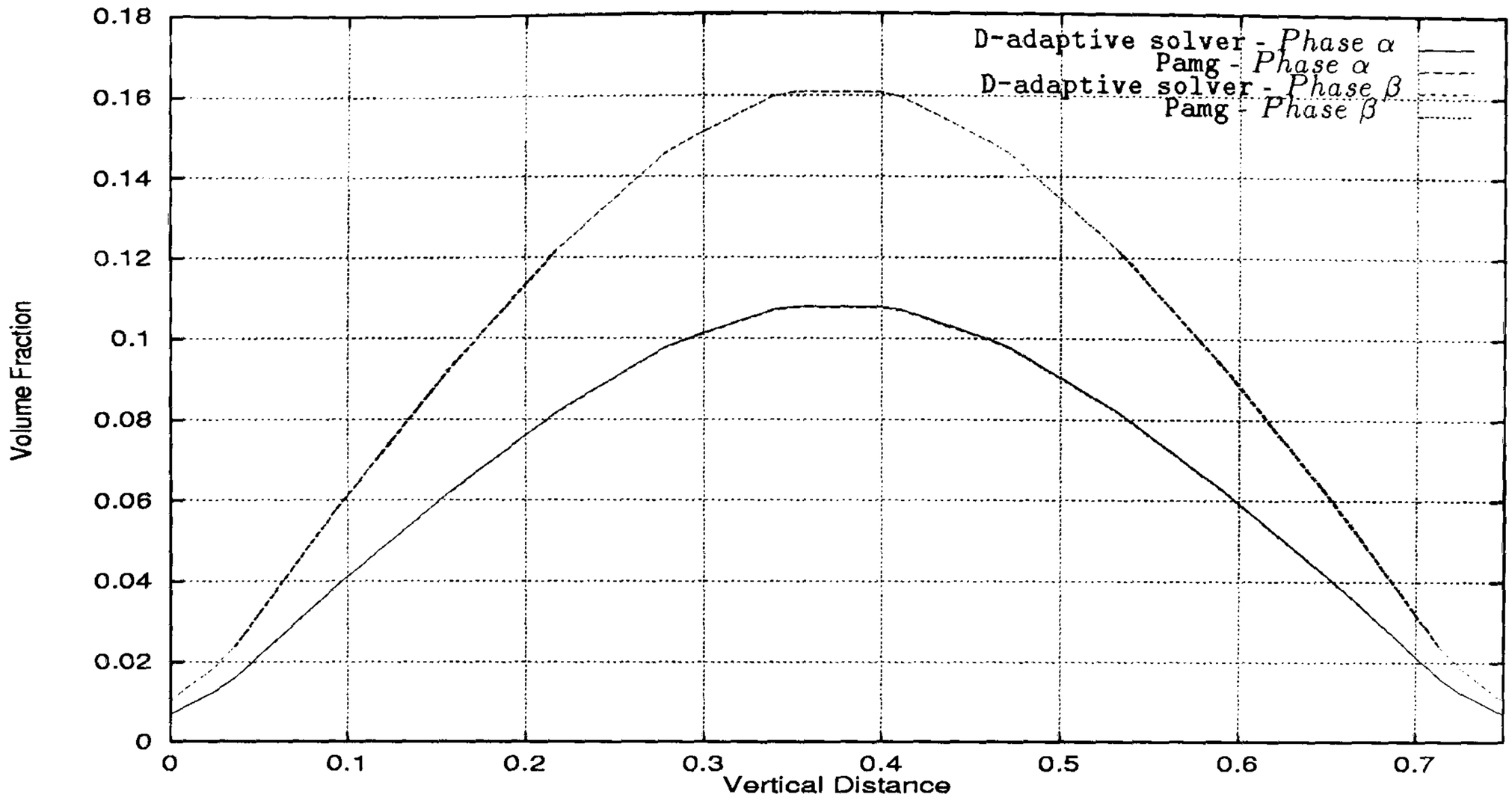


Figure 6.29: Two-phase flow through a Contraction (Case I) - comparison between the horizontal velocity profiles along the line $x = 7.125$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the excellent agreement between the parabolic profiles produced by both solvers.

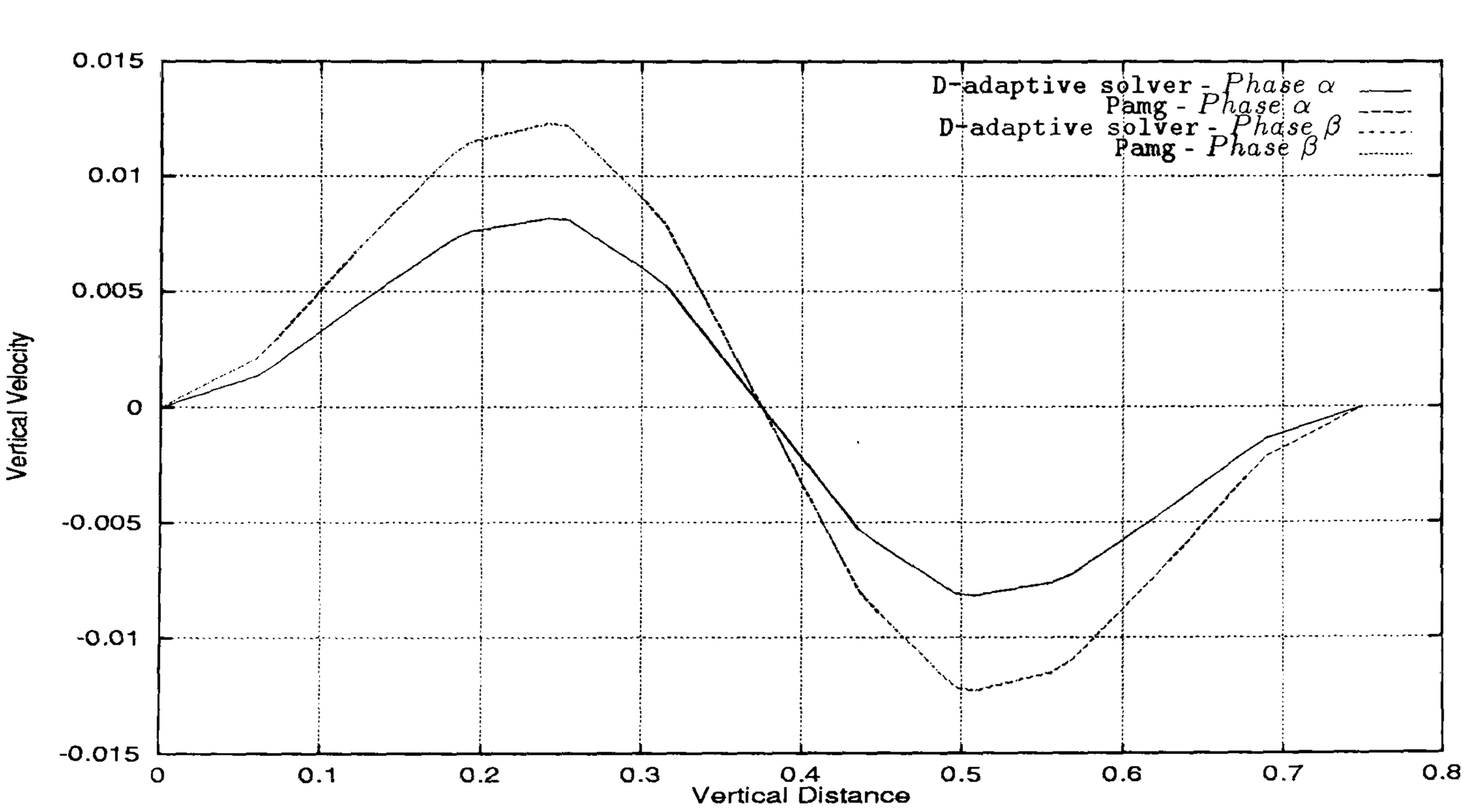


Figure 6.30: Two-phase flow through a Contraction (Case I) - comparison between the vertical velocity profiles along the line $x = 7.125$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the symmetry of the vertical velocities varying from positive to negative values. It can be observed the excellent agreement between the two solvers.

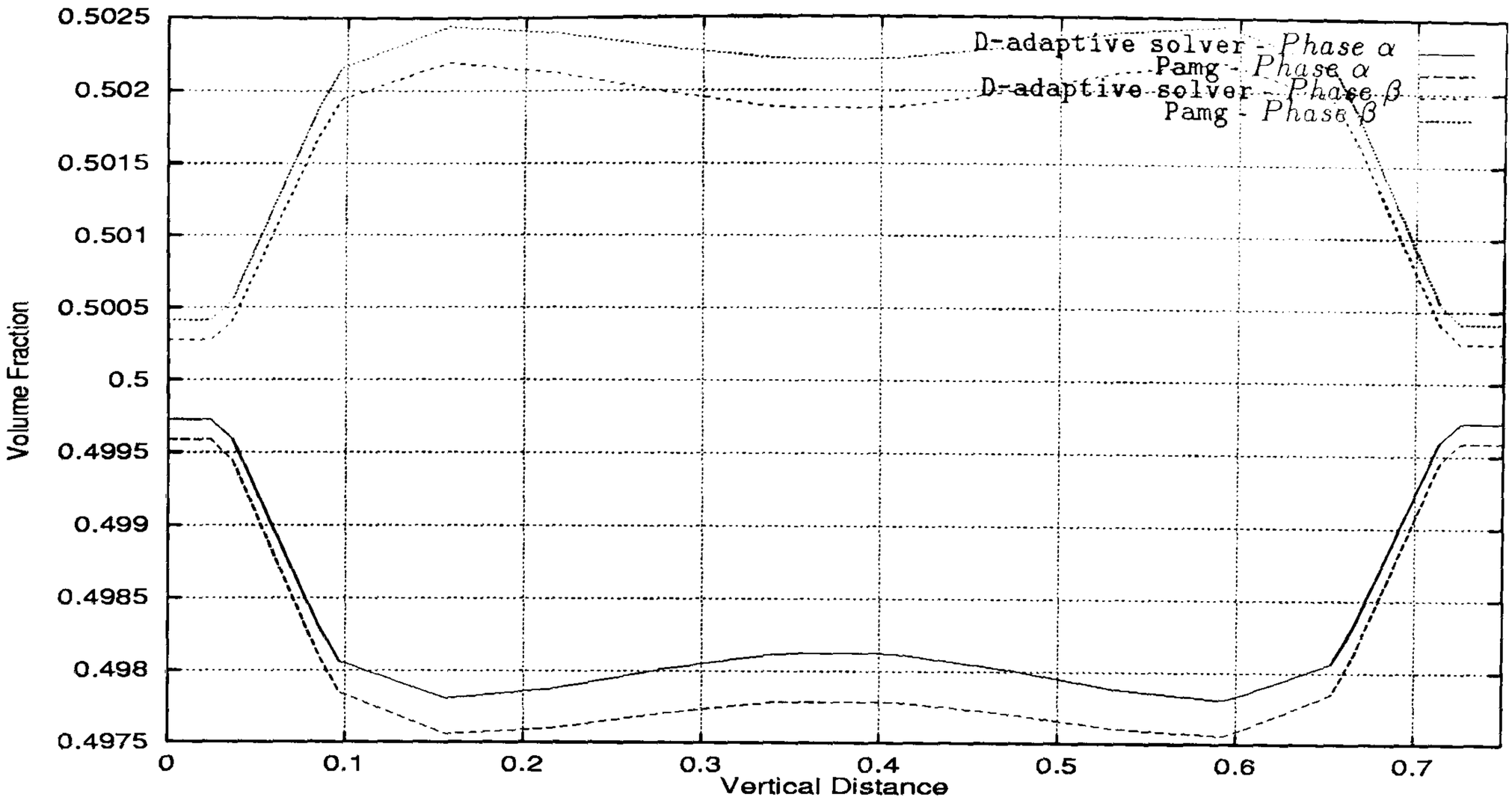


Figure 6.31: Two-phase flow through a Contraction (Case I) - comparison between the volume fractions profiles along the line $x = 7.125$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be observed the good agreement between the two solvers. Note the very small scale of the volume fraction axis.

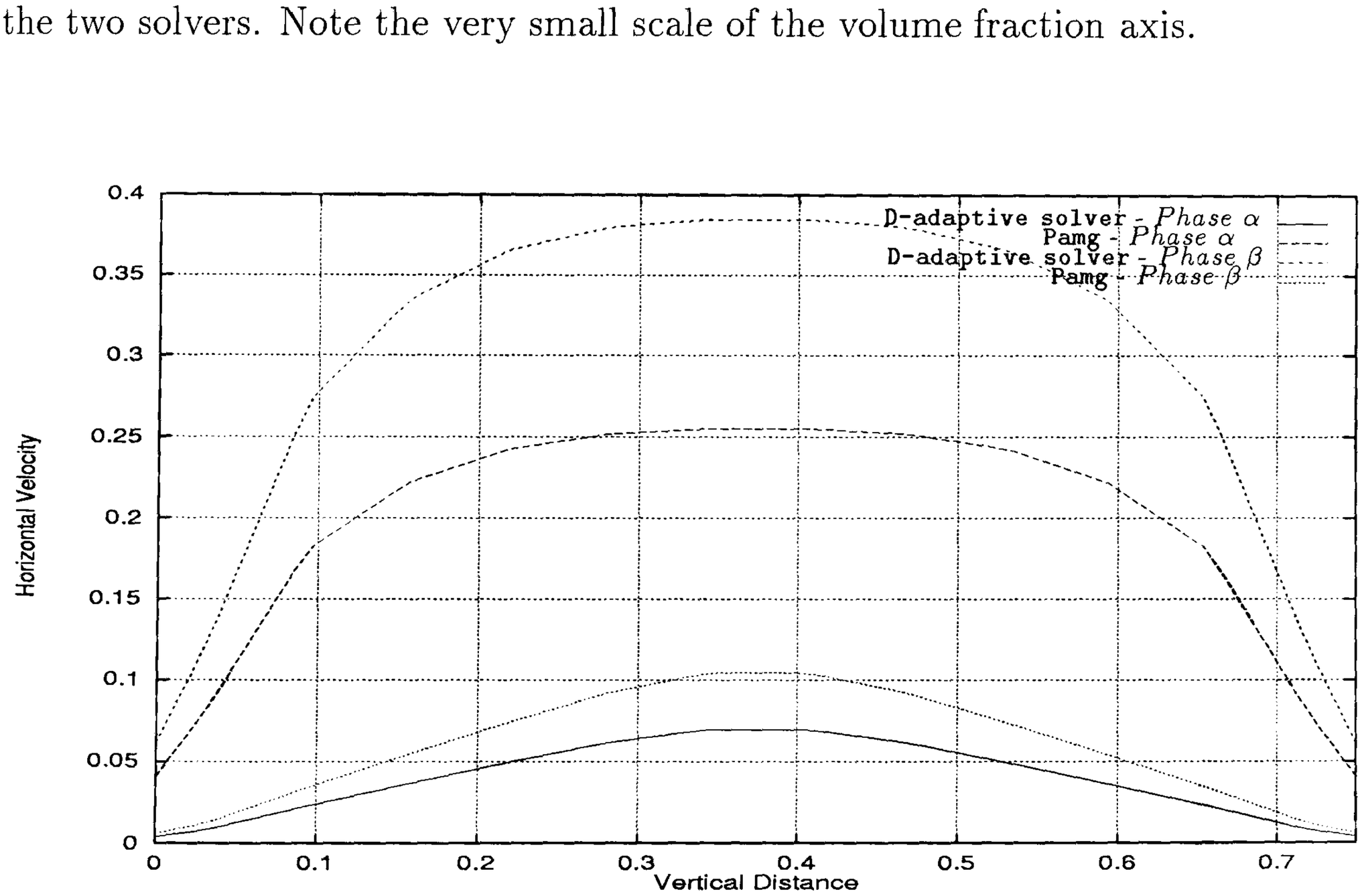


Figure 6.32: Two-phase flow through a Contraction (Case I) - comparison between the horizontal velocity profiles along the line $x = 8.250$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the same parabolic pattern in both solvers.

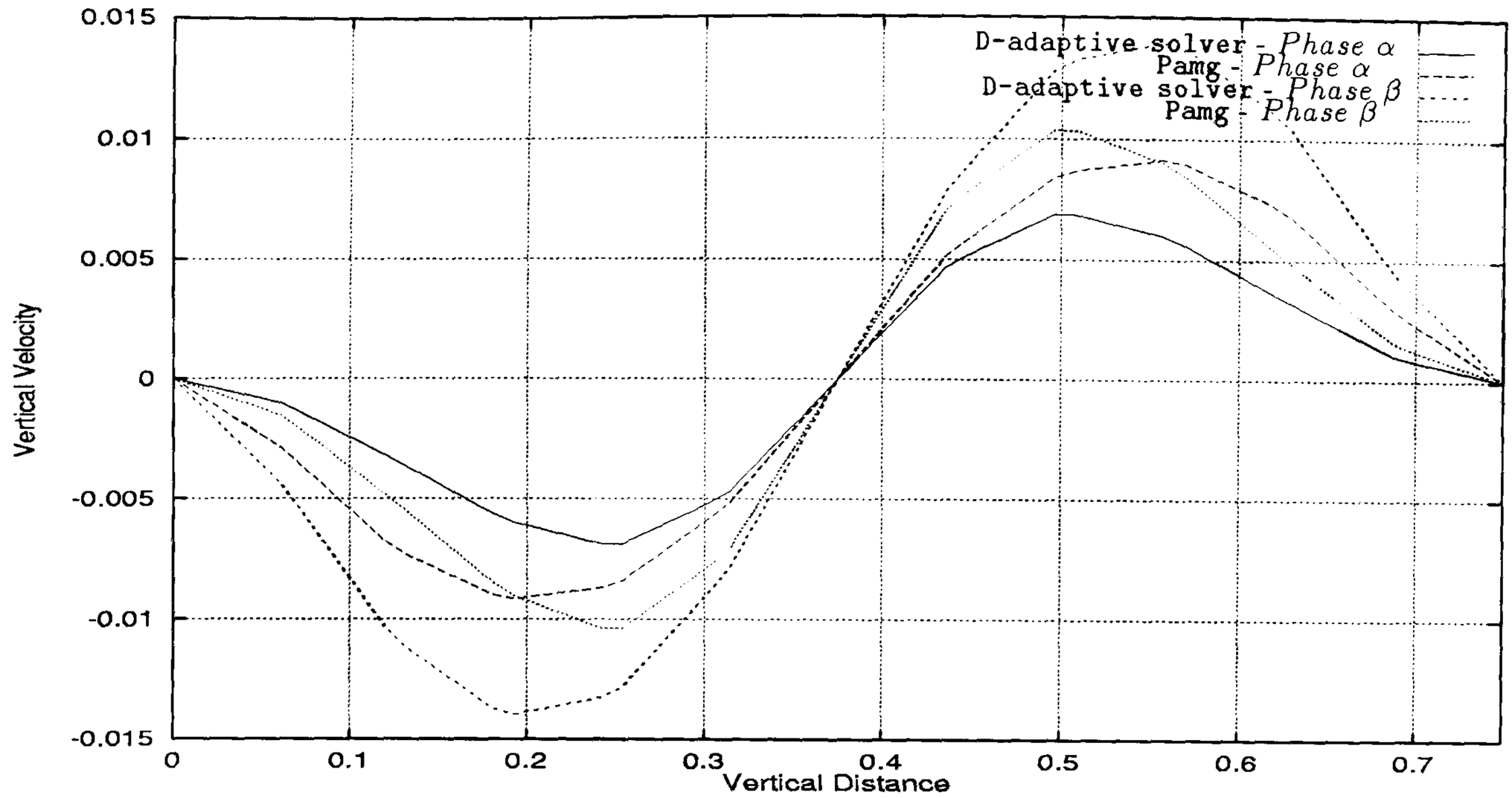


Figure 6.33: Two-phase flow through a Contraction (Case I) - comparison between the vertical velocity profiles along the line $x = 8.250$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be observed the variation of the vertical velocities from negative to positive values. The pattern exhibited by both solvers is similar.

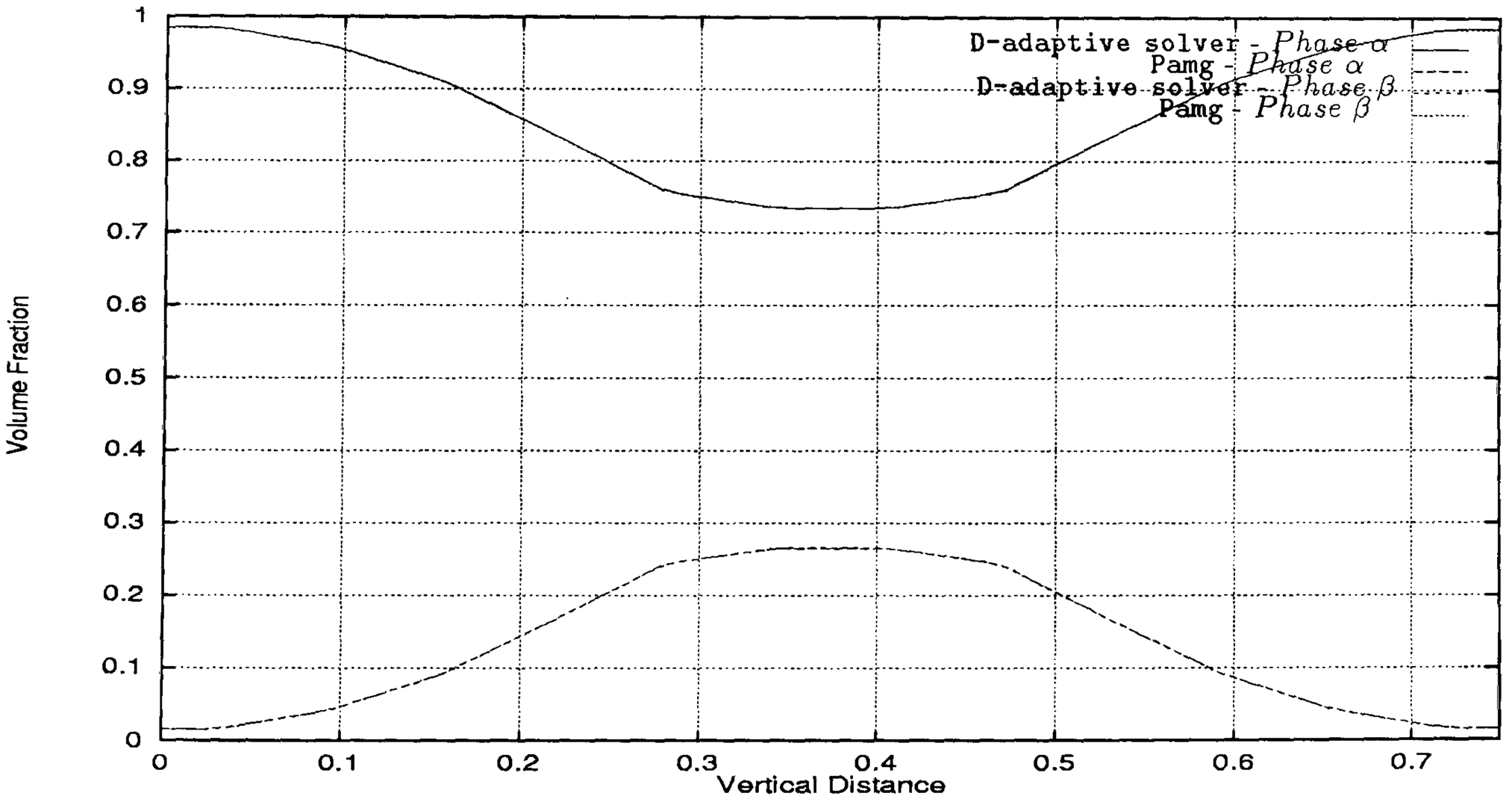


Figure 6.34: Two-phase flow through a Contraction (Case I) - comparison between the volume fraction profiles along the line $x = 8.250$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the excellent agreement between the two solvers over all the vertical distance.

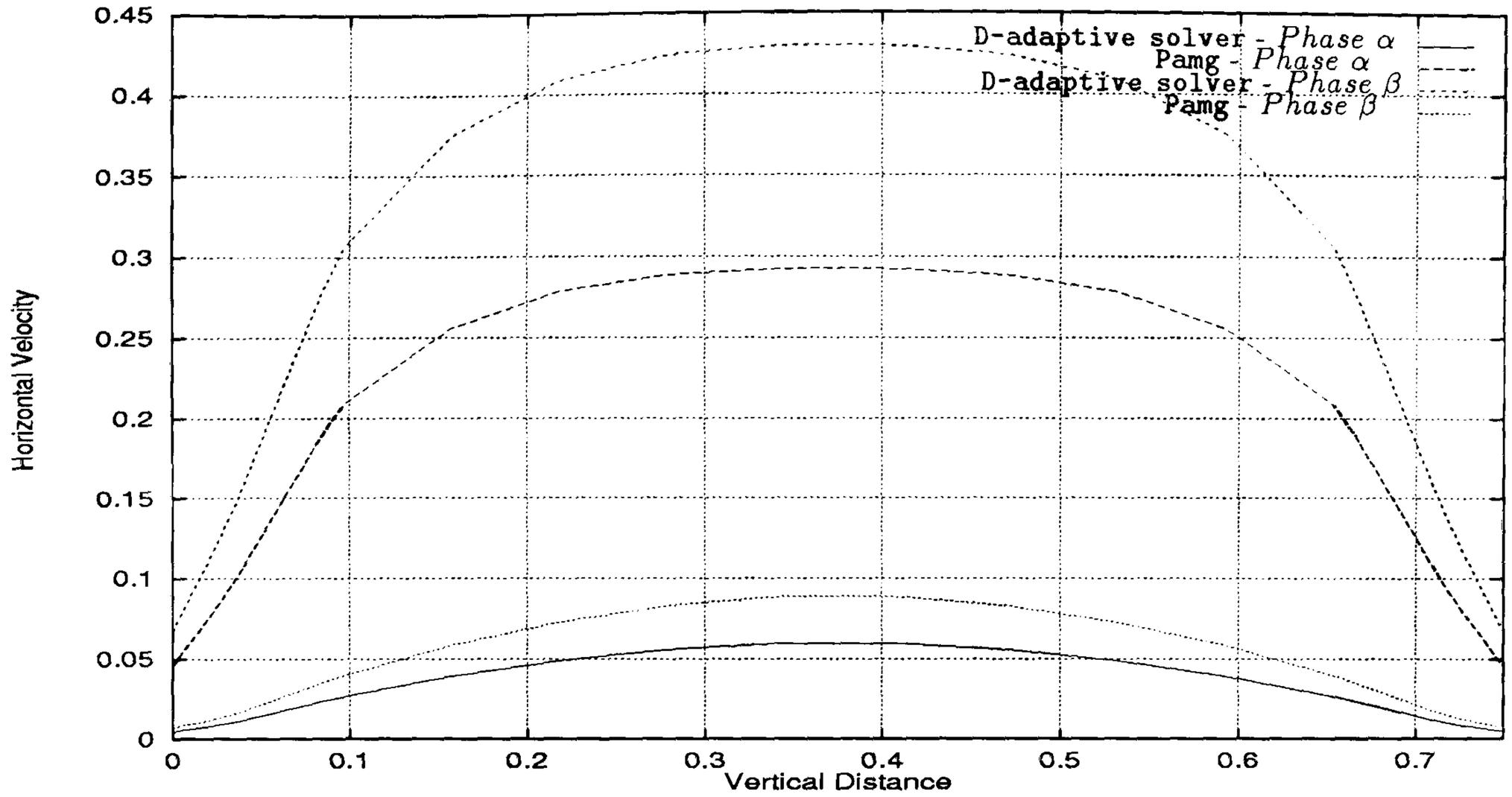


Figure 6.35: Two-phase flow through a Contraction (Case I) - comparison between the horizontal velocity profile along the line $x = 9.350$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the parabolic profile of the horizontal velocities for both solvers.

vertical velocities and volume fractions along the line $x = 9.375$, which has been defined as the second interface line for this experiment-in this case between the two-dimensional region and the one-dimensional one. It is visible the parabolic shape of the horizontal velocity. The vertical velocities as can be seen in Figure 6.36 are as we expected null. The volume fractions produced by the two solvers show a very good agreement as can be observed in Figure 6.37.

Finally in Figures 6.38 and 6.39 we present the values of the horizontal velocities and volume fractions at the outlet. The straight lines corresponding to the horizontal velocities and volume fractions produced by the `D-adaptive solver` mean that naturally in the one-dimensional region we only have a value and the concept of vertical distance is not applicable. However this was the manner that we find to present the results in the outlet line.

Another important issue is quite naturally the question of the performance of the two solvers. In this scope it is very important to measure the computational cost of each one. The `D-adaptive solver` due to how is constructed is necessarily much more economic in terms of memory storage. However, its main advantage is on the CPU-time spent to simulate a two-phase fluid flow particular in a complex geometry.

In Table 6.1 we present the CPU-time spent in the simulation of this two-phase flow by the `D-adaptive solver` and the `transient Pamg Multi-phase`. The `D-adaptive solver` is more than 10 times faster than the `transient Pamg`

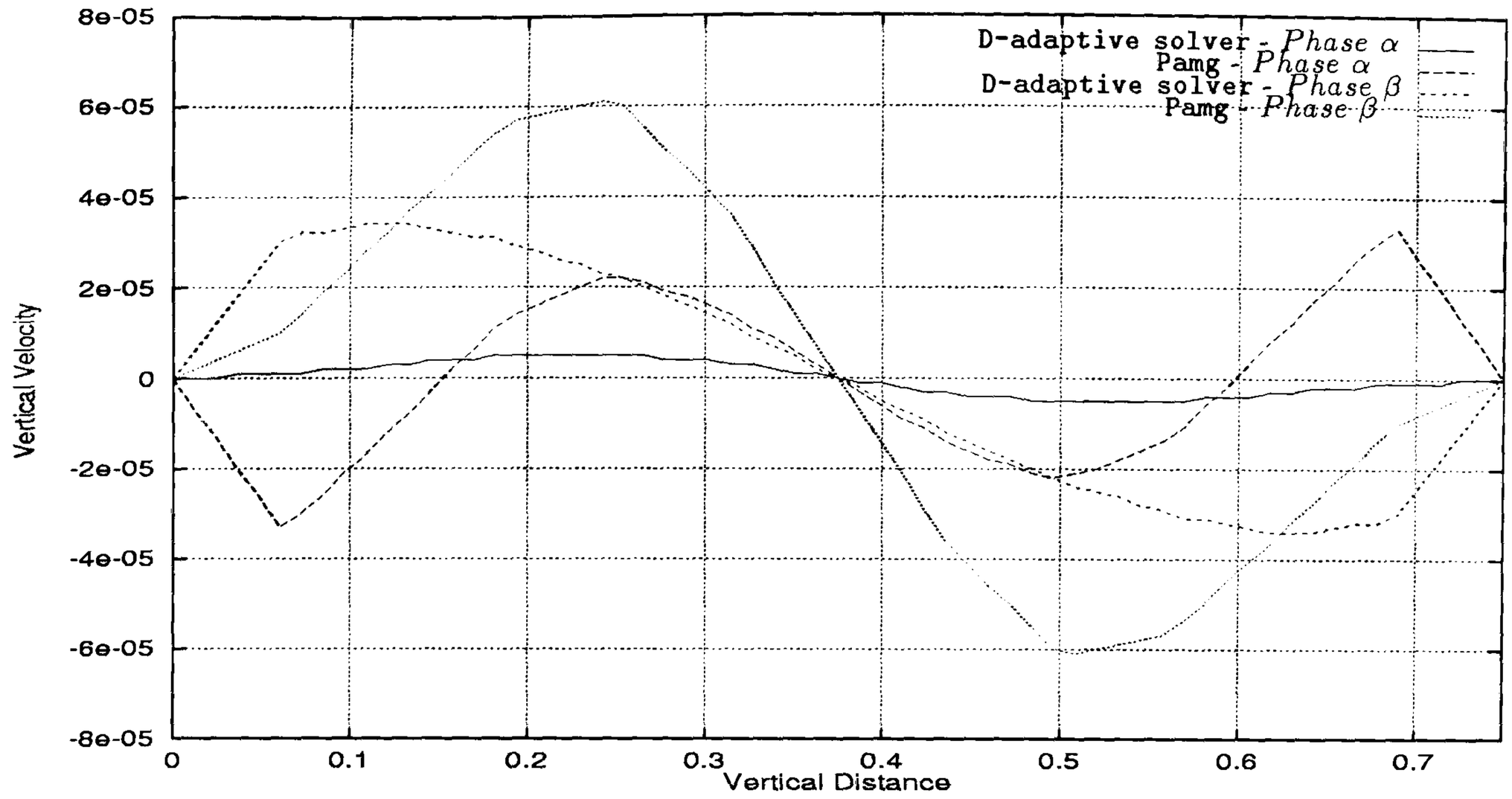


Figure 6.36: Two-phase flow through a Contraction (Case I) - comparison between the vertical velocity profiles along the line $x = 9.350$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It is visible that the vertical velocites are null for both solvers. Note the scale of the vertical velocity axis.

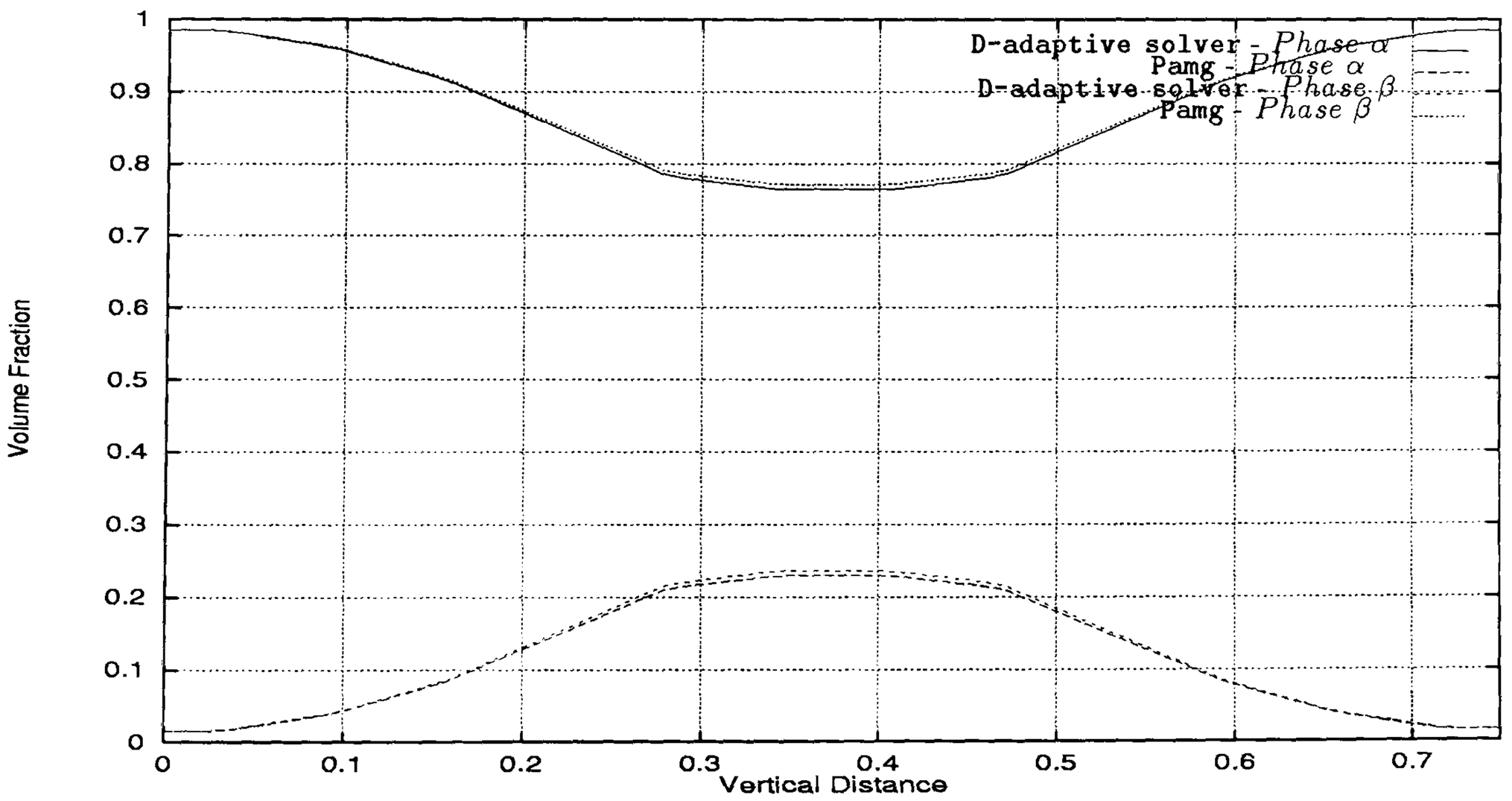


Figure 6.37: Two-phase flow through a Contraction (Case I) - comparison between the volume fraction profiles along the line $x = 9.350$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the same pattern of the volume fractions profiles for both solvers.

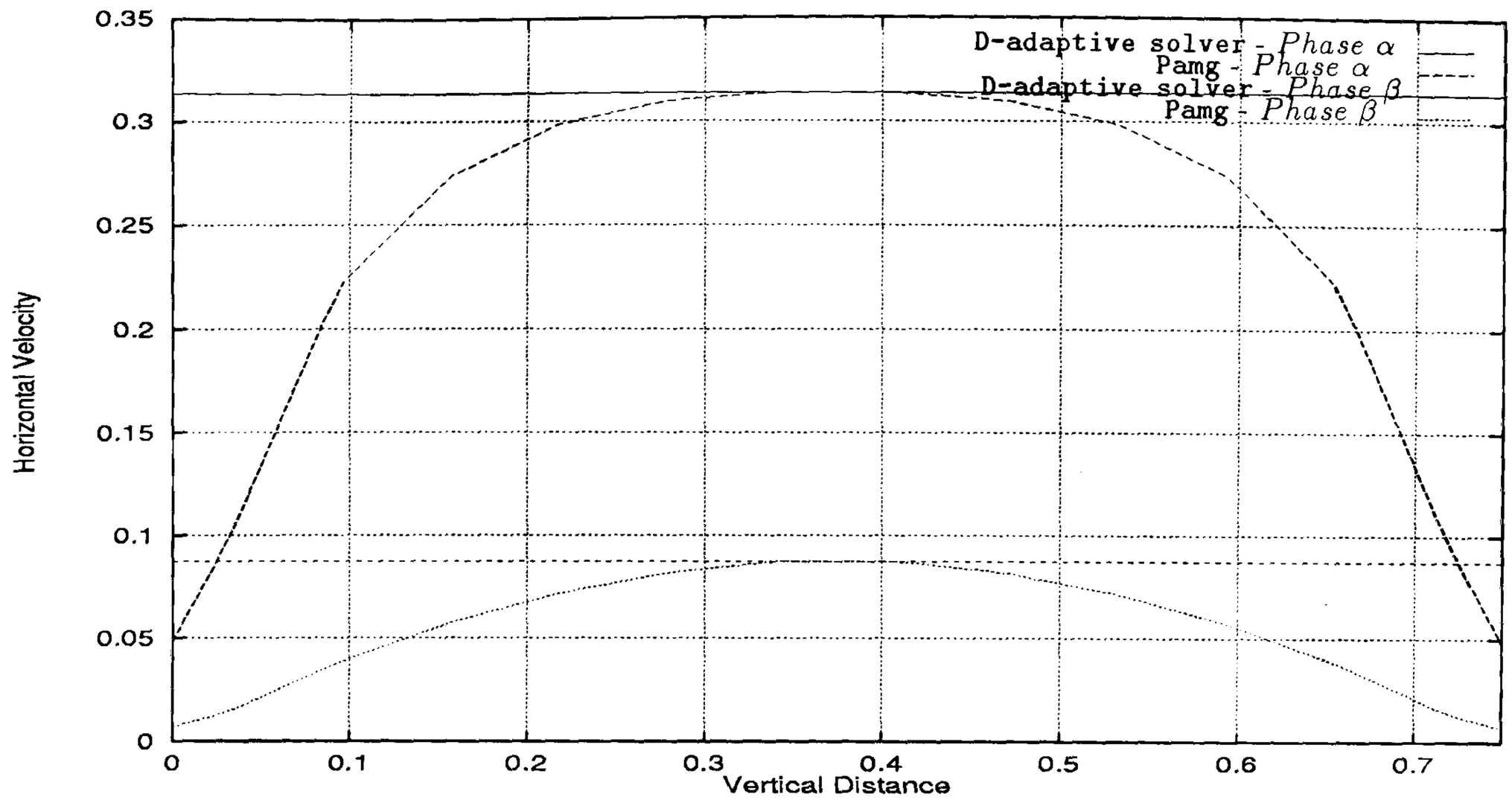


Figure 6.38: Two-phase flow through a Contraction (Case I) - comparison between the horizontal velocity profiles along the line $x = 15.350$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the parabolic profiles produced by `transient Pamg Multi-phase` code. The straight lines correspondent to the `D-adaptive solver` in this one-dimensional region.

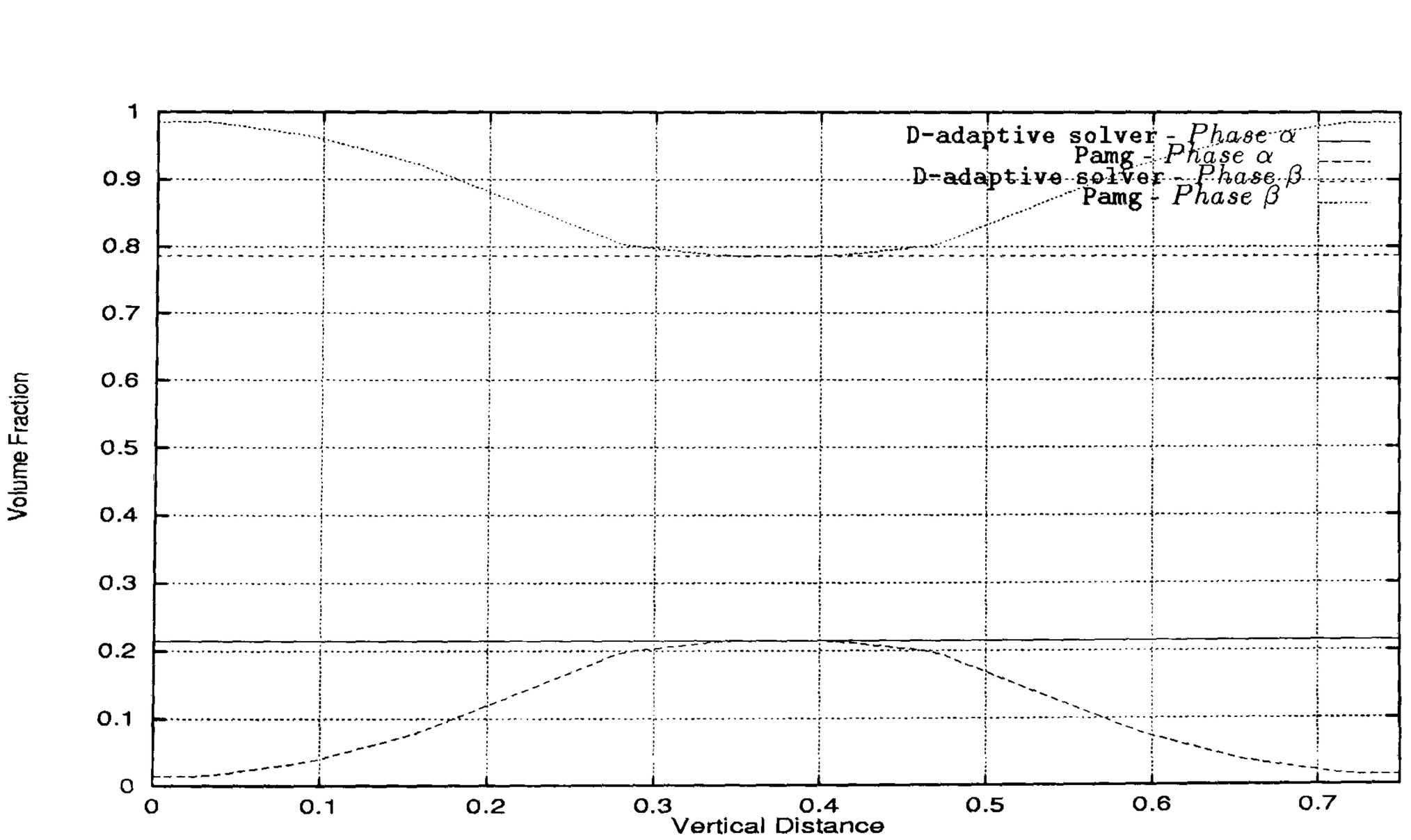


Figure 6.39: Two-phase flow through a Contraction (Case I) - comparison between the volume fraction profiles along the line $x = 15.350$ for `D-adaptive solver` and `transient Pamg Multi-phase`. The straight lines correspond to the `D-adaptive solver` in this one-dimensional region.

`Multi-phase` for this particular test case.

Two-Phase Flow	`D-adaptive solver`	`Pamg`
First One-dimensional domain	215 sec	
Two-dimensional domain	808 sec	
Second One-dimensional domain	224 sec	
Overall domain	1247 sec	17634 sec

Table 6.1: CPU-time spent by the `D-adaptive solver` and the `transient Pamg Multi-phase` in the simulation of the contraction two-phase flow (Case I)

Another experiment with the same domain has been outlined. The main objective of this experiment was to complement the preceding one with the case of having exactly the same density for both phases. Furthermore we could verify if some of the discrepancies observed in the results of the preceding experiment still remain or have some variation.

Hence the physical properties chosen for this case are:

Two-Phase Flow	Reynolds Number	Viscosity	Density
Phase α	15	0.01	1.0
Phase β	15	0.005	1.0

The boundary conditions in the inlet are:

- Velocities $u(x) = 0.2$ for $Phase\ \alpha$ and $u(x) = 0.4$ for $Phase\ \beta$
- Volume fractions $r(x) = 0.5$ for both phases

Naturally to guarantee that the relation between the velocity of $Phase\ \alpha$ with the velocity of $Phase\ \beta$ is kept the coefficients chosen for the Hydrodynamic closure law (see Equation (2.17)) have to be updated in order to keep this relation between the velocity of the two phases.

The results along the line $y = 0.375$ are presented in Figures 6.40 through 6.42. The horizontal velocity profiles, which can be seen in Figure 6.40, exhibit a better agreement than the experiment before. However in the first one-dimensional region while the `D-adaptive solver` maintains the initial velocity, `transient Pamg Multi-phase` equalises the velocity of the two phases. The little jumps that we see when changing from the two-dimensional region to the second one-dimensional one in the `D-adaptive solver` profile are due to the interpolation errors in the interface itf_2. Figure 6.41 shows the volume fractions profiles along the line $y = 0.375$. In the first one-dimensional region while the `D-adaptive solver` maintains the volume fractions equal to 0.5 the `transient Pamg multi-phase` modifies these values and consequently the profiles do not agree. This different behaviour is due to different initial guess for the volume fractions. Figure 6.42 shows the pressure drop along the line $y = 0.375$ for the `D-adaptive solver` and `transient Pamg Multi-phase`. This graph is very similar to the equivalent one produced in the above experiment. The pressure drop in the contraction region shows the same shape but in the other two one-dimensional regions the straight lines are not parallel.

In Figures 6.43 and 6.44 we present the results for the horizontal velocities and volume fractions along the line $x = 6.0$. The differences in the behaviour of the horizontal velocities and on the volume fractions profiles which have been verified on the line $y = 0.375$ are observed here.

Figures 6.45, 6.46 and 6.47 show the results for the horizontal velocities, vertical velocities and volume fractions along the line $x = 7.125$. In this line we can see

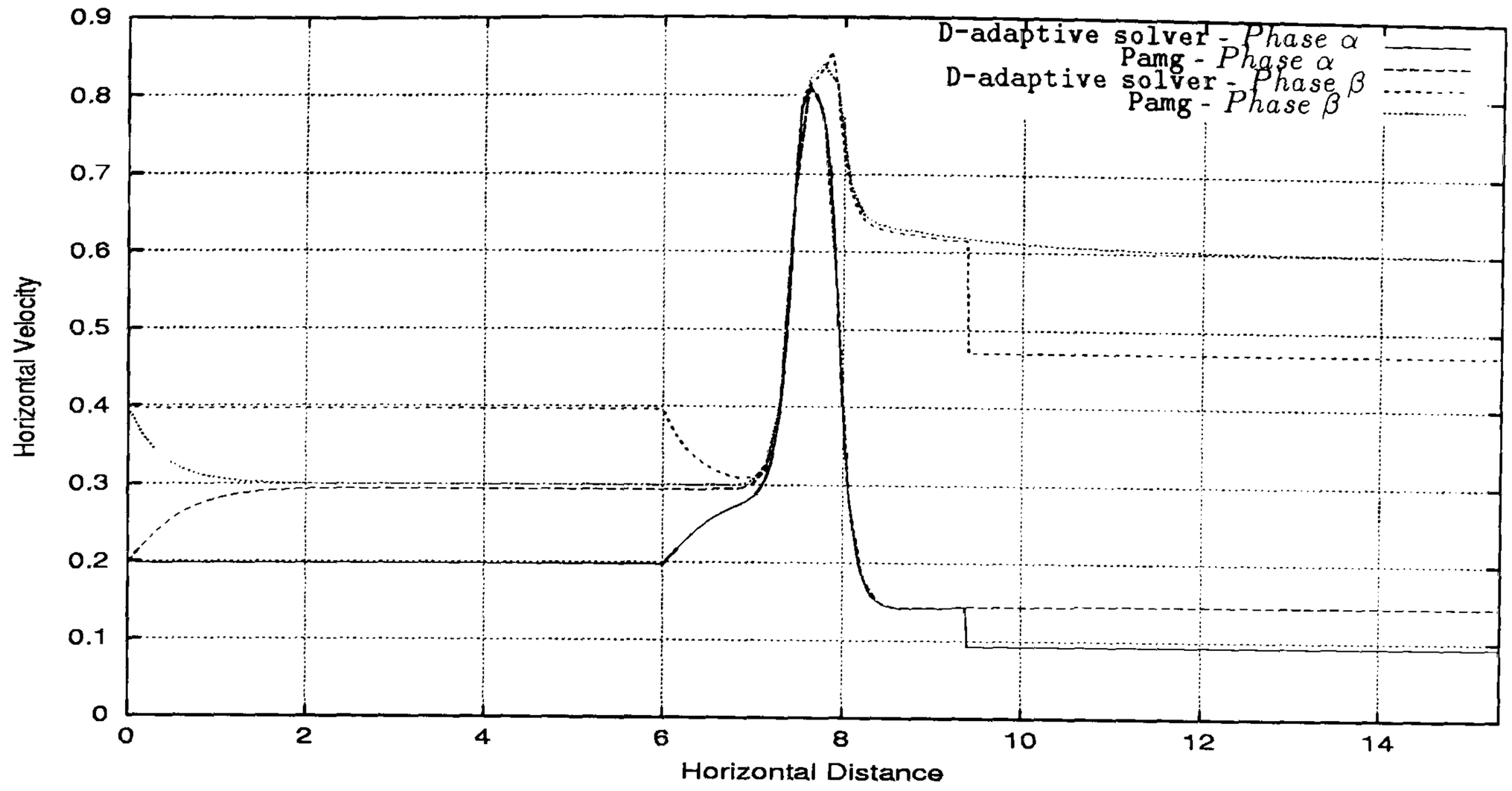


Figure 6.40: Two-phase flow through a Contraction (Case II) - comparison between the horizontal velocity along the line $y = 0.375$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen that `transient Pamg Multi-phase` solver equalises the velocities of both phases in the beggining of the horizontal distance. The jumps in the `D-adaptive solver` results are at the two-dimensional/one-dimensional interface.

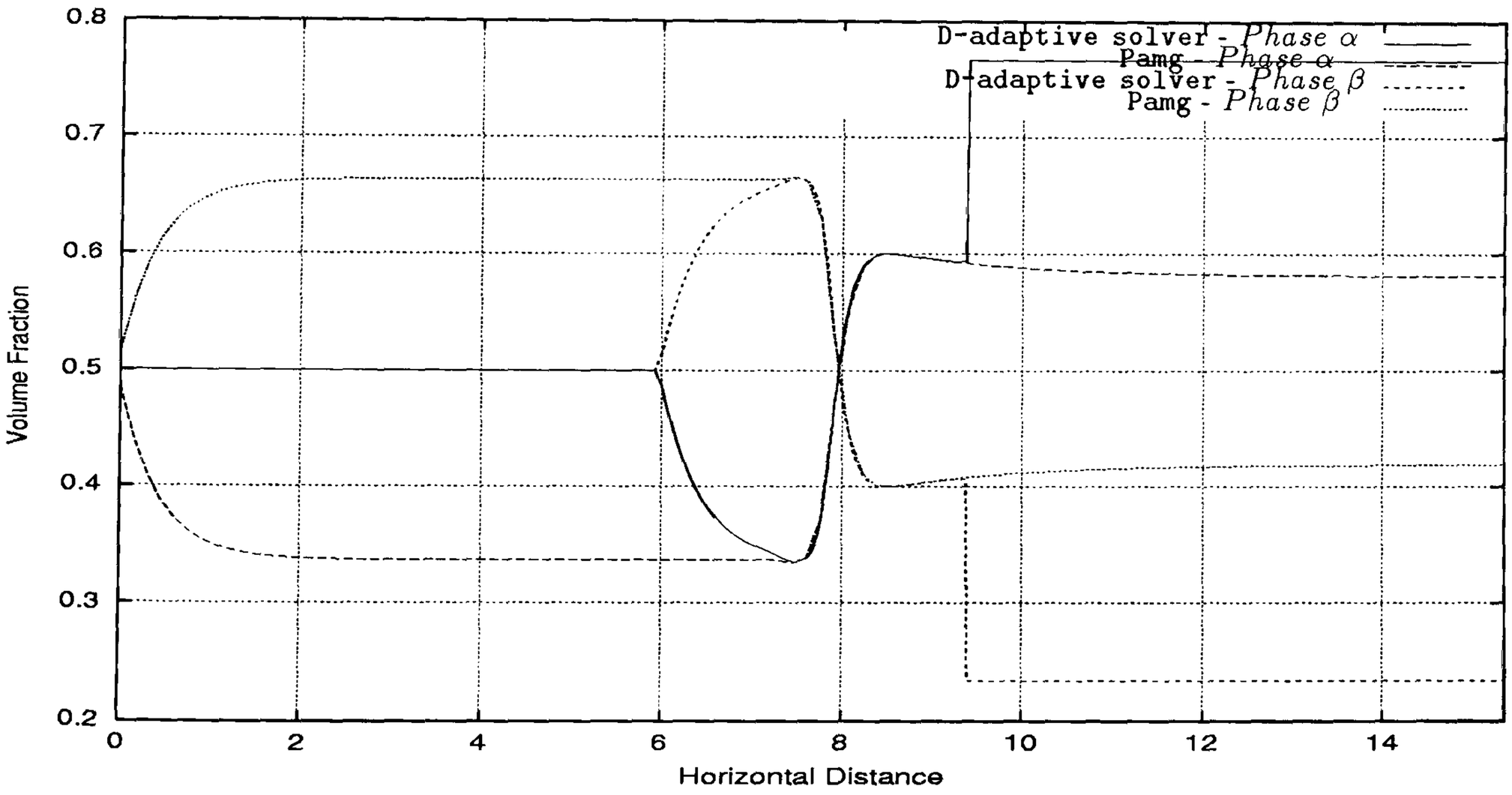


Figure 6.41: Two-phase flow through a Contraction (Case II) - comparison between the volume fraction profile along the line $y = 0.375$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen that `transient Pamg Multi-phase` solver modifies the initial value of the volume fractions while the `D-adaptive solver` maintains the initial value in the first one-dimensional region.

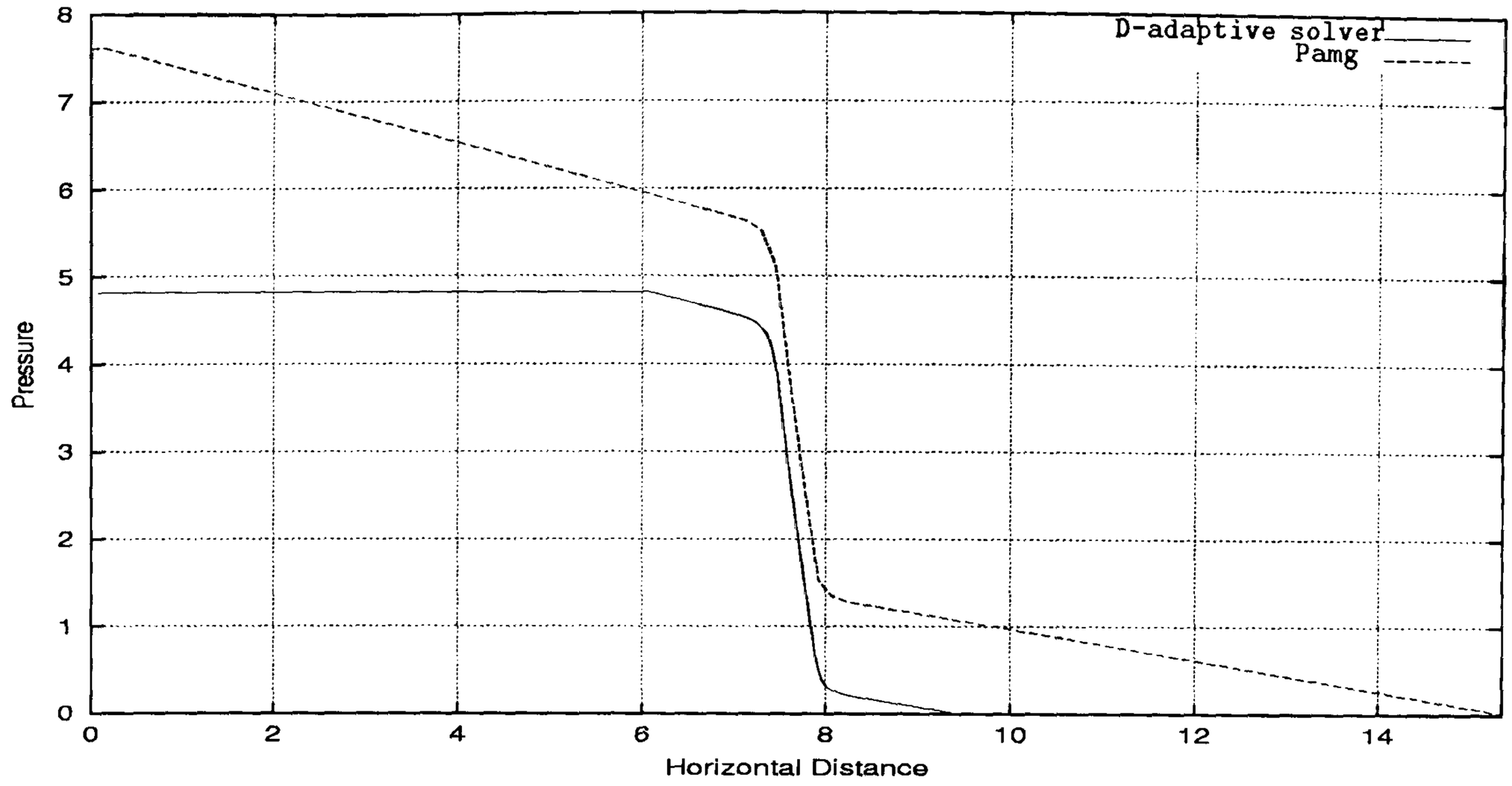


Figure 6.42: Two-phase flow through a Contraction (Case II) - comparison between the pressure drop profile along the line $y = 0.375$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be observed the same profile in the contraction region. In the one-dimensional regions the lines produced by both solvers are not parallel.

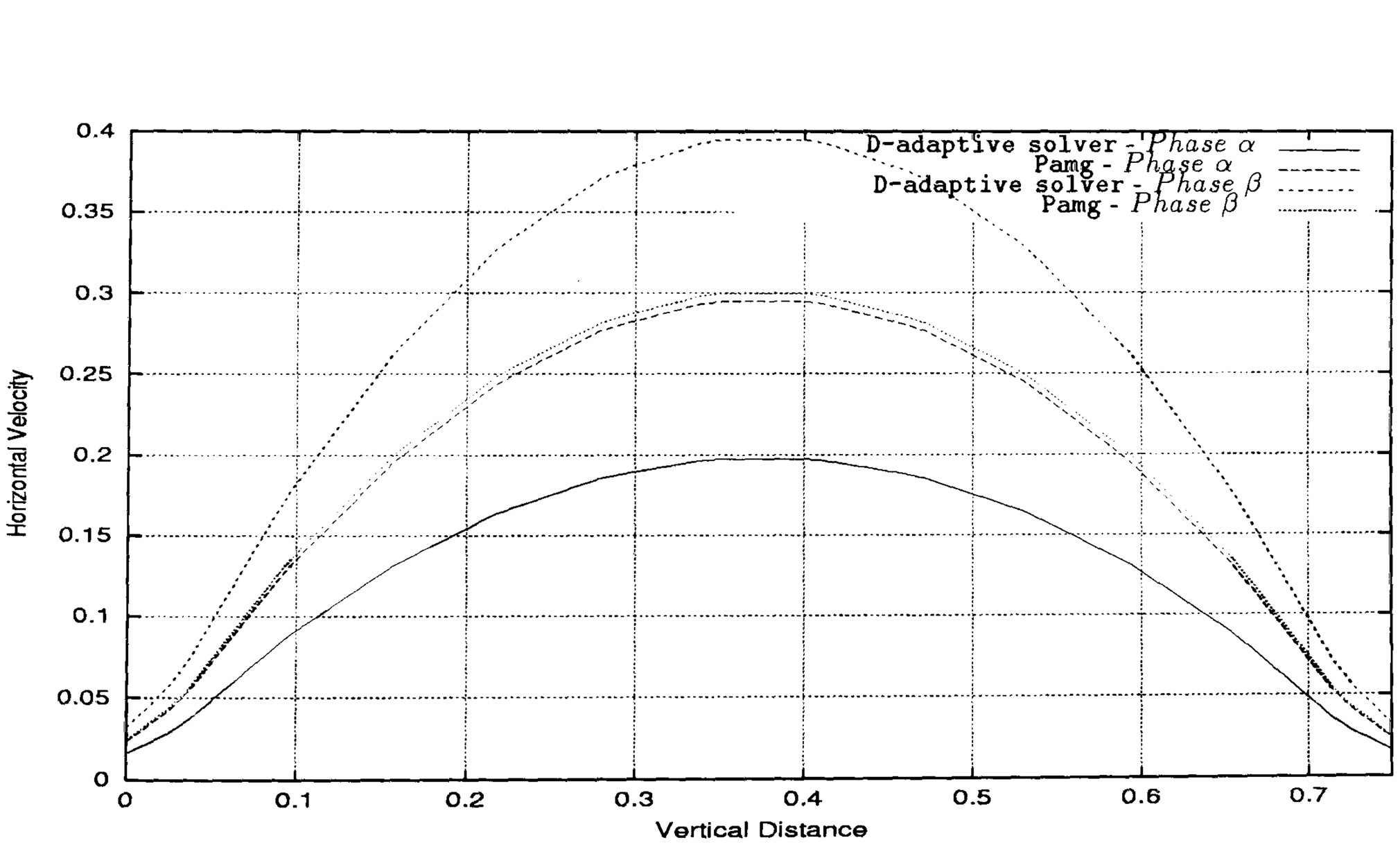


Figure 6.43: Two-phase flow through a Contraction (Case II) - comparison between the horizontal velocity profiles along the line $x = 6.0$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be observed the parabolic velocity profile for both solvers.

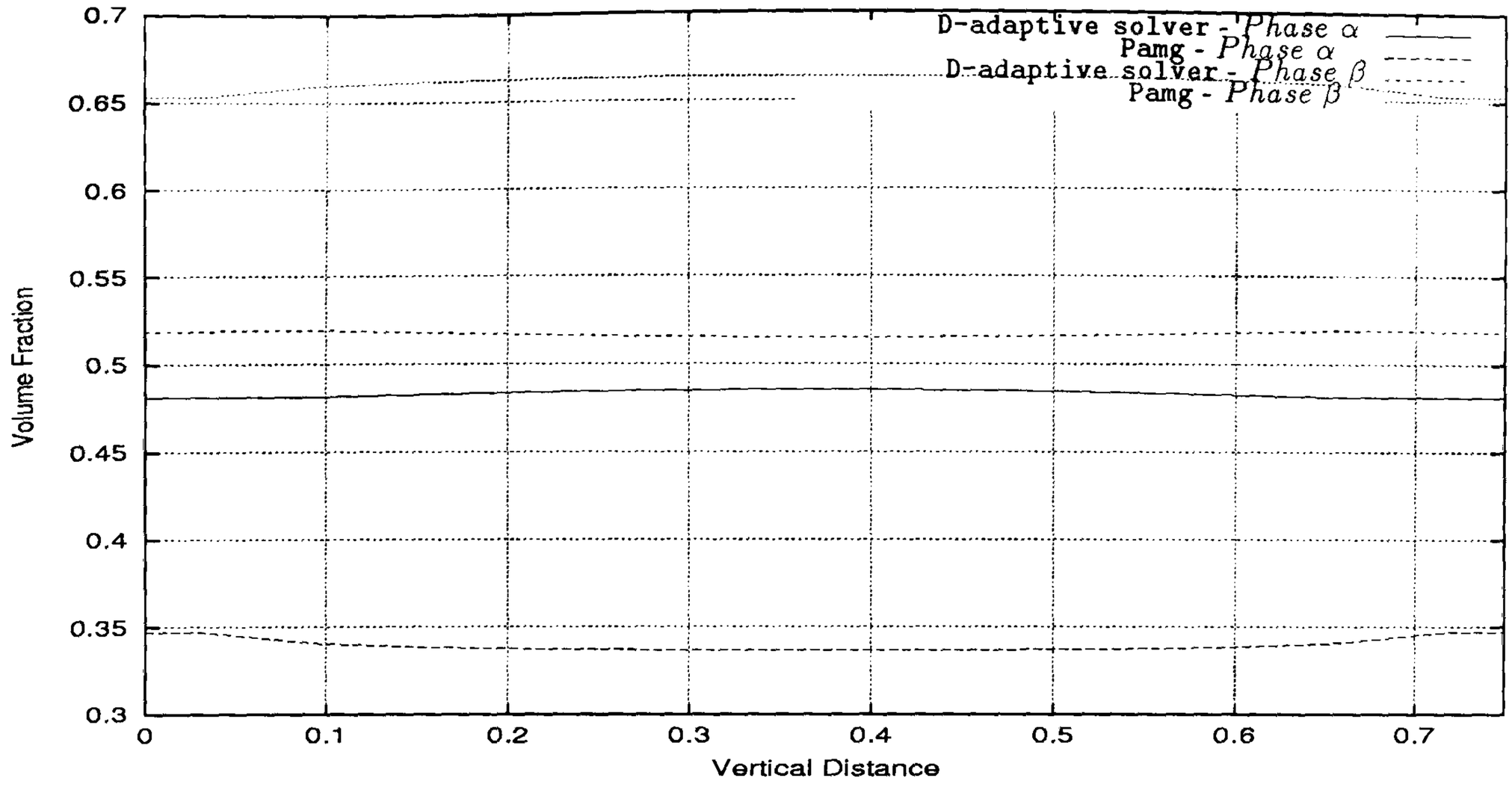


Figure 6.44: Two-phase flow through a Contraction (Case II) - comparison between the volume fraction profiles along the line $x = 6.0$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be observed that at this first interface region the `D-adaptive solver` keeps the initial condition for the volume fractions while `transient Pamg Multi-phase` solver changes substantially the initial guess.

good agreement between all the curves. In Figure 6.45 we see that the parabolic profiles of the horizontal velocity are all very close and in Figure 6.46 the vertical velocity also match very effectively. Moreover the volume fractions (see Figure 6.47) have values almost constant along the vertical distance and the agreement between the results of the two solvers is generally good for both phases.

In Figures 6.48, 6.49 and 6.50 we can see the results for the horizontal velocities, vertical velocities and volume fractions along the line $x = 8.250$ produced by both solvers. The horizontal velocity profiles, which are plotted in Figure 6.48 agree in a very effective manner and exhibit the parabolic expected profile. In this difficult region the vertical velocities are effective which can be seen in Figure 6.49. The volume fraction profiles match almost perfectly (see Figure 6.50).

In Figures 6.51 and 6.52 the results for the horizontal velocities and volume fractions along the line $x = 9.350$ produced by both solvers are presented. The horizontal velocity profiles produced by `D-adaptive solver` and `transient Pamg Multi-phase` exhibit an excelent agreement and the expected parabolic shape. The *Phase* β is faster than *Phase* α which is basically due to its smaller viscosity. The vertical velocities in this line are null. The volume fractions produced by both solvers clearly agree as can be seen in Figure 6.52.

Finally in Figures 6.53 and 6.54 we present the values of the horizontal velocities

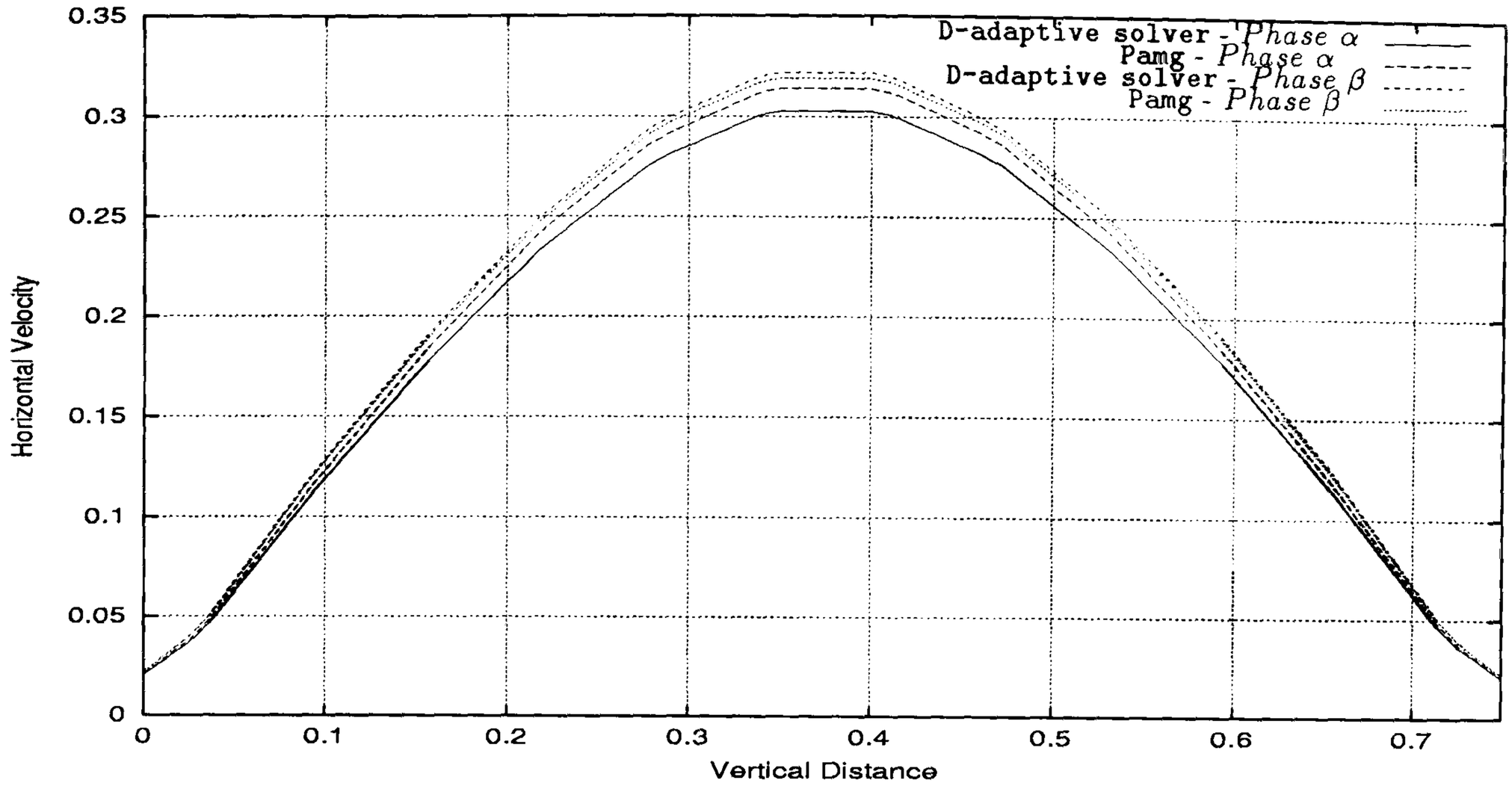


Figure 6.45: Two-phase flow through a Contraction (Case II) - comparison between the horizontal velocity profiles along the line $x = 7.125$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the good agreement between the parabolic profiles produced by both solvers.

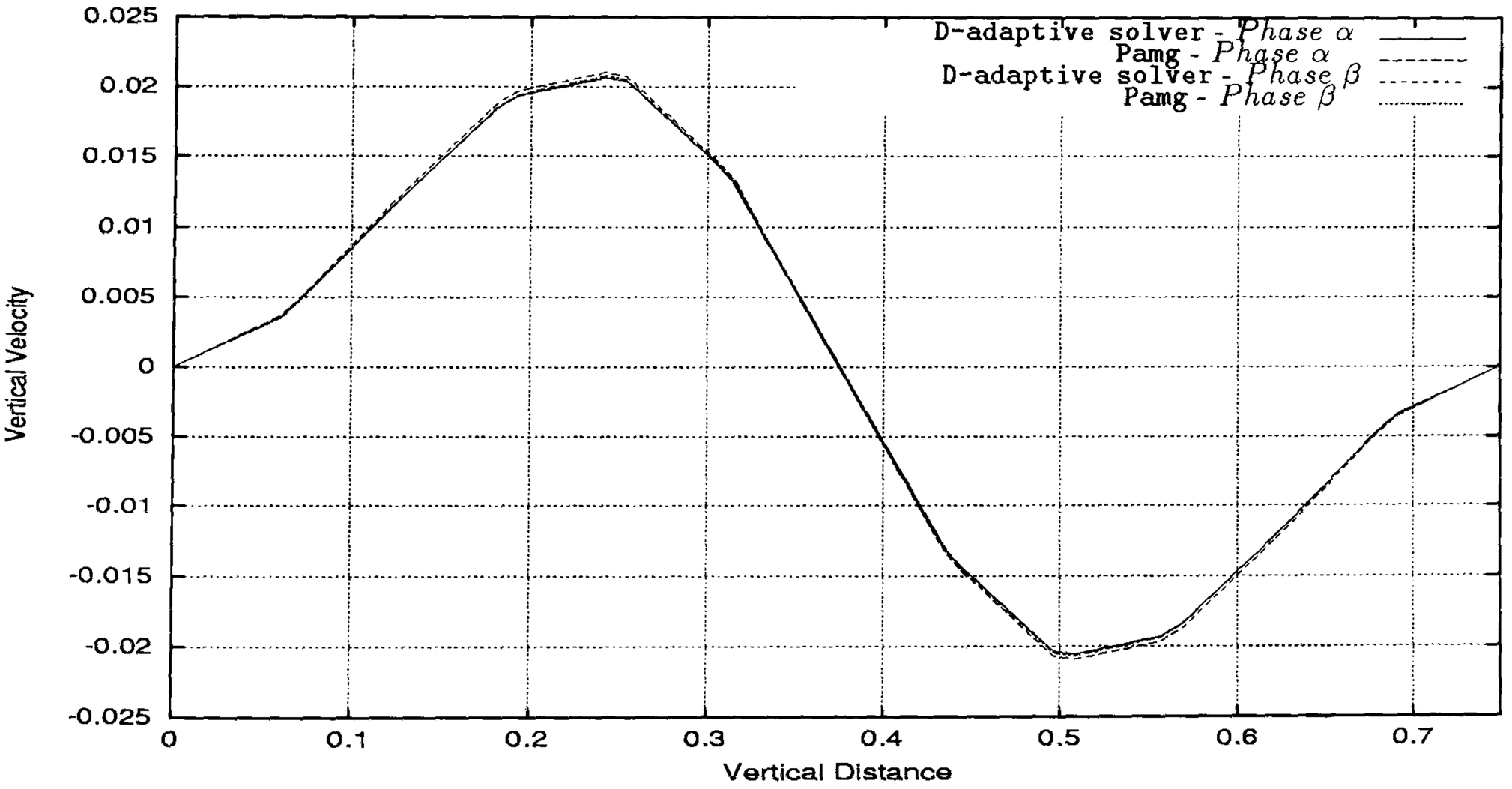


Figure 6.46: Two-phase flow through a Contraction (Case II) - comparison between the vertical velocity profile along the line $x = 7.125$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be observed that the vertical velocities agree very well over all the vertical distance.

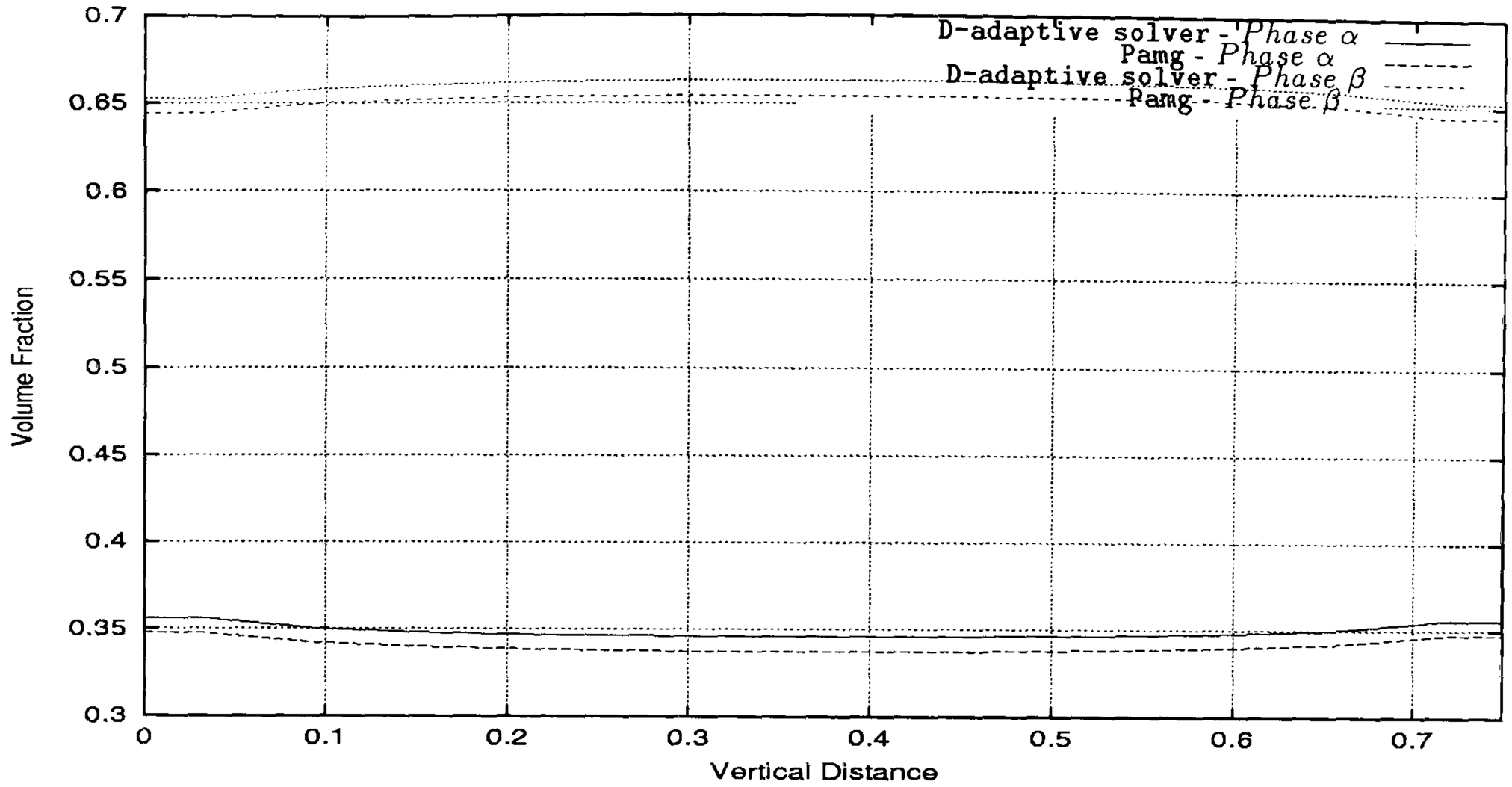


Figure 6.47: Two-phase flow through a Contraction (Case II) - comparison between the volume fraction profiles along the line $x = 7.125$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen that the volume fractions are almost constant along the vertical distance. Good agreement between the two solvers can be observed for both phases.

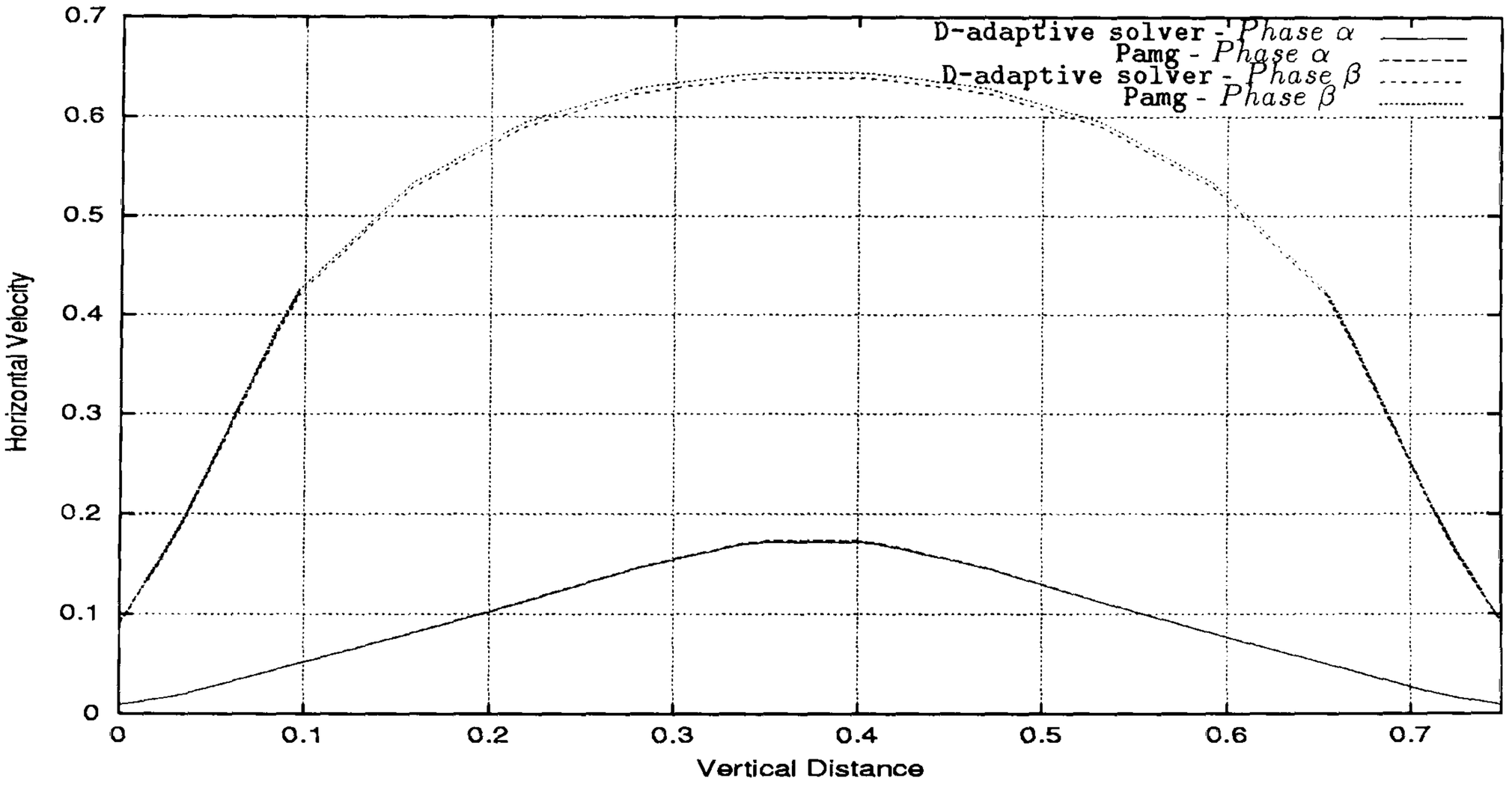


Figure 6.48: Two-phase flow through a Contraction (Case II) - comparison between the horizontal velocity profile along the line $x = 8.250$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the good agreement between the parabolic profiles produced by both solvers.

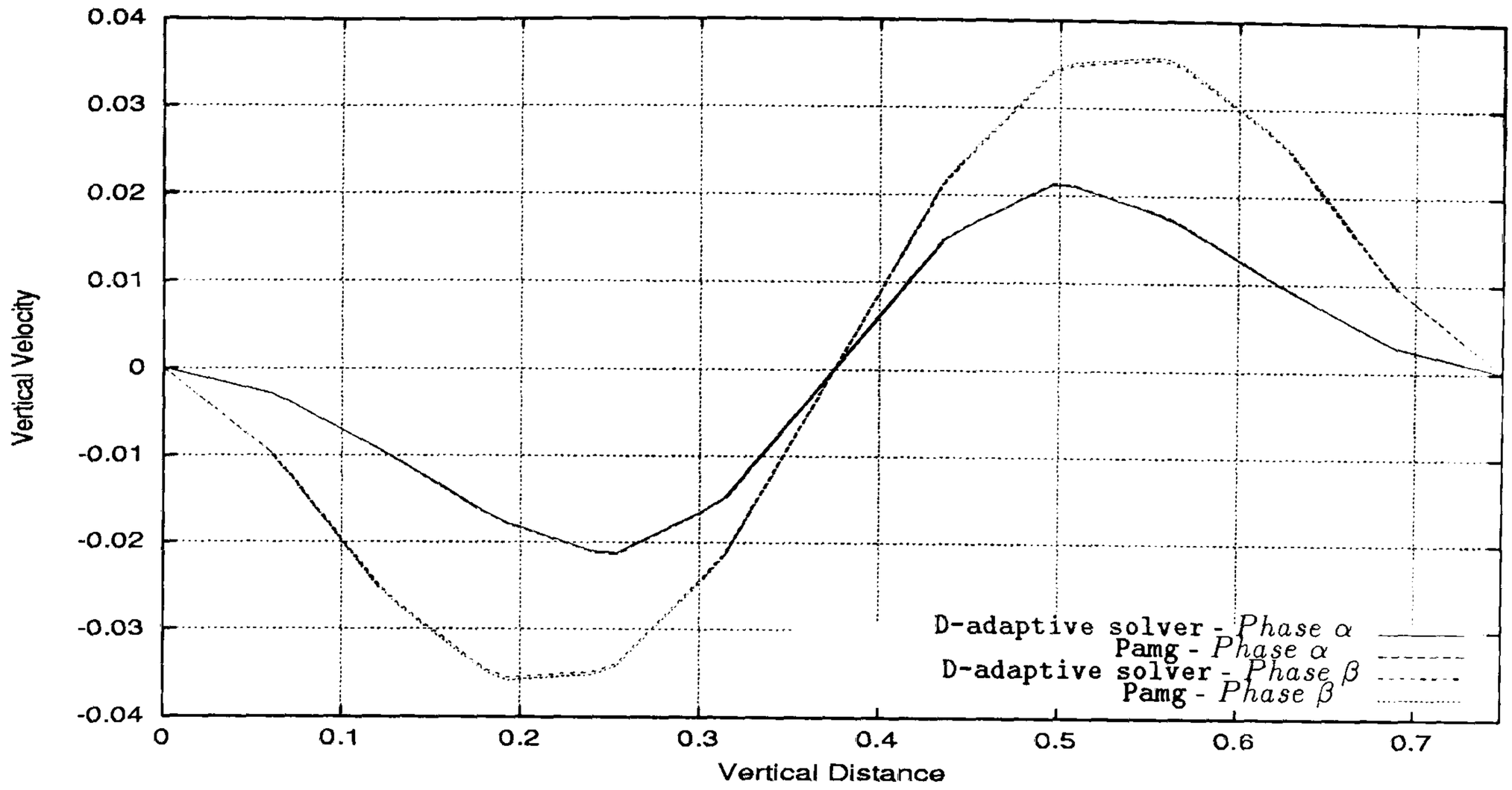


Figure 6.49: Two-phase flow through a Contraction (Case II) - comparison between the vertical velocity profile along the line $x = 8.250$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be observed the good agreement between the vertical velocities produced by both solvers over all the entire vertical distance.

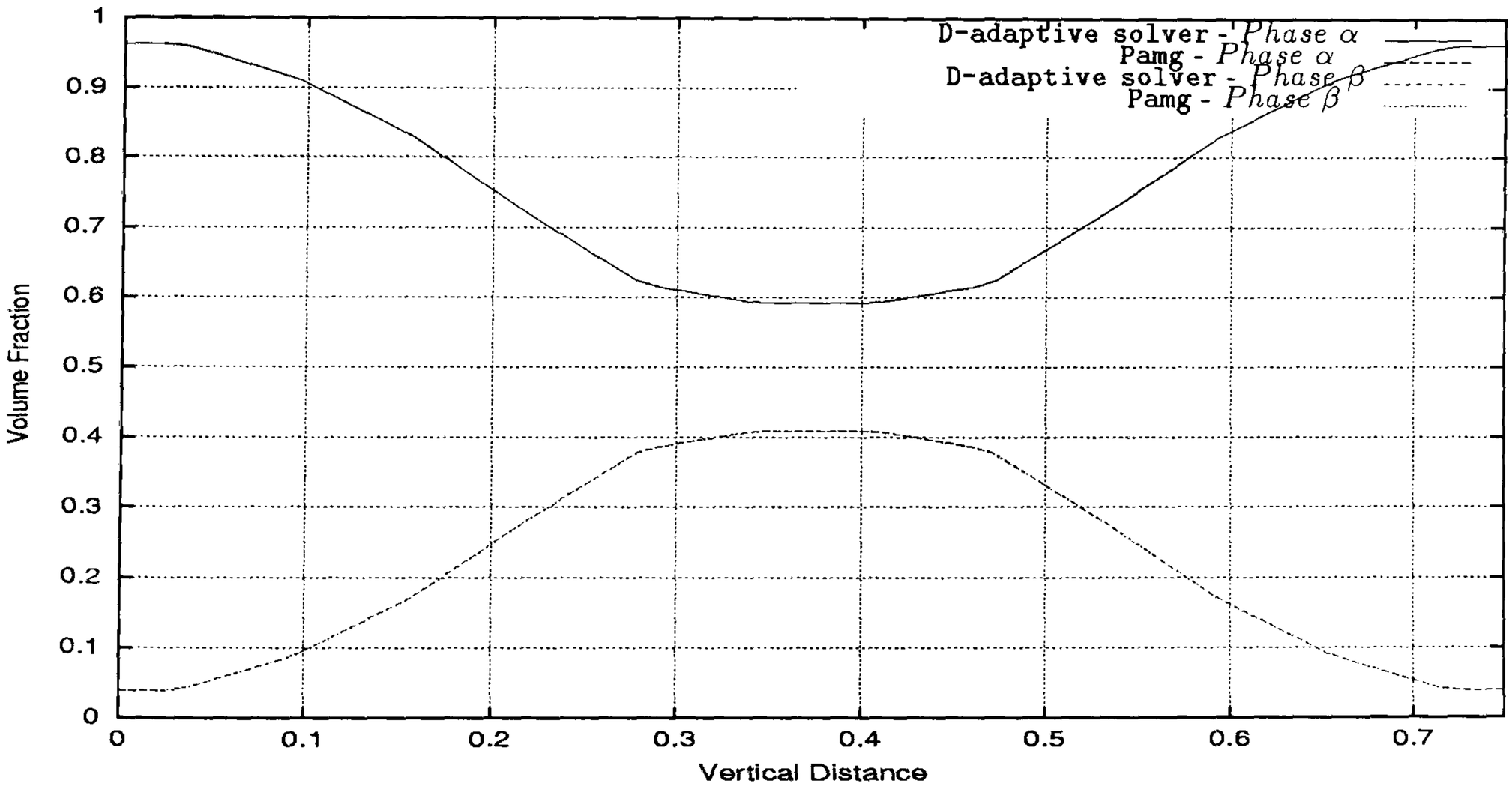


Figure 6.50: Two-phase flow through a Contraction (Case II) - comparison between the volume fraction profile along the line $x = 8.250$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be verified the excellent agreement between the volume fraction profiles produced by both solvers.

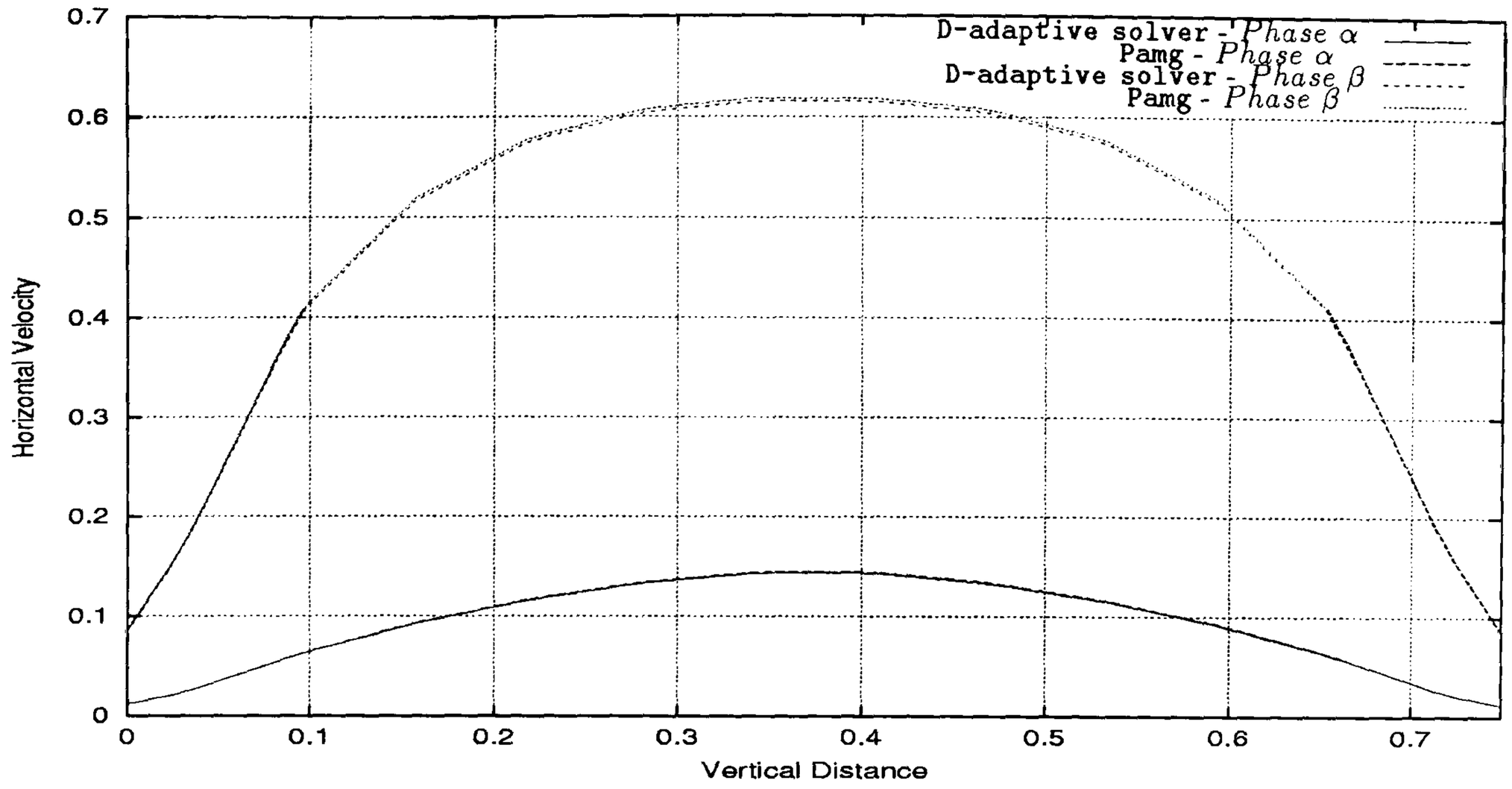


Figure 6.51: Two-phase flow through a Contraction (Case II) - comparison between the horizontal velocity profile along the line $x = 9.350$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the good agreement between the parabolic velocity profiles produced by both solvers.

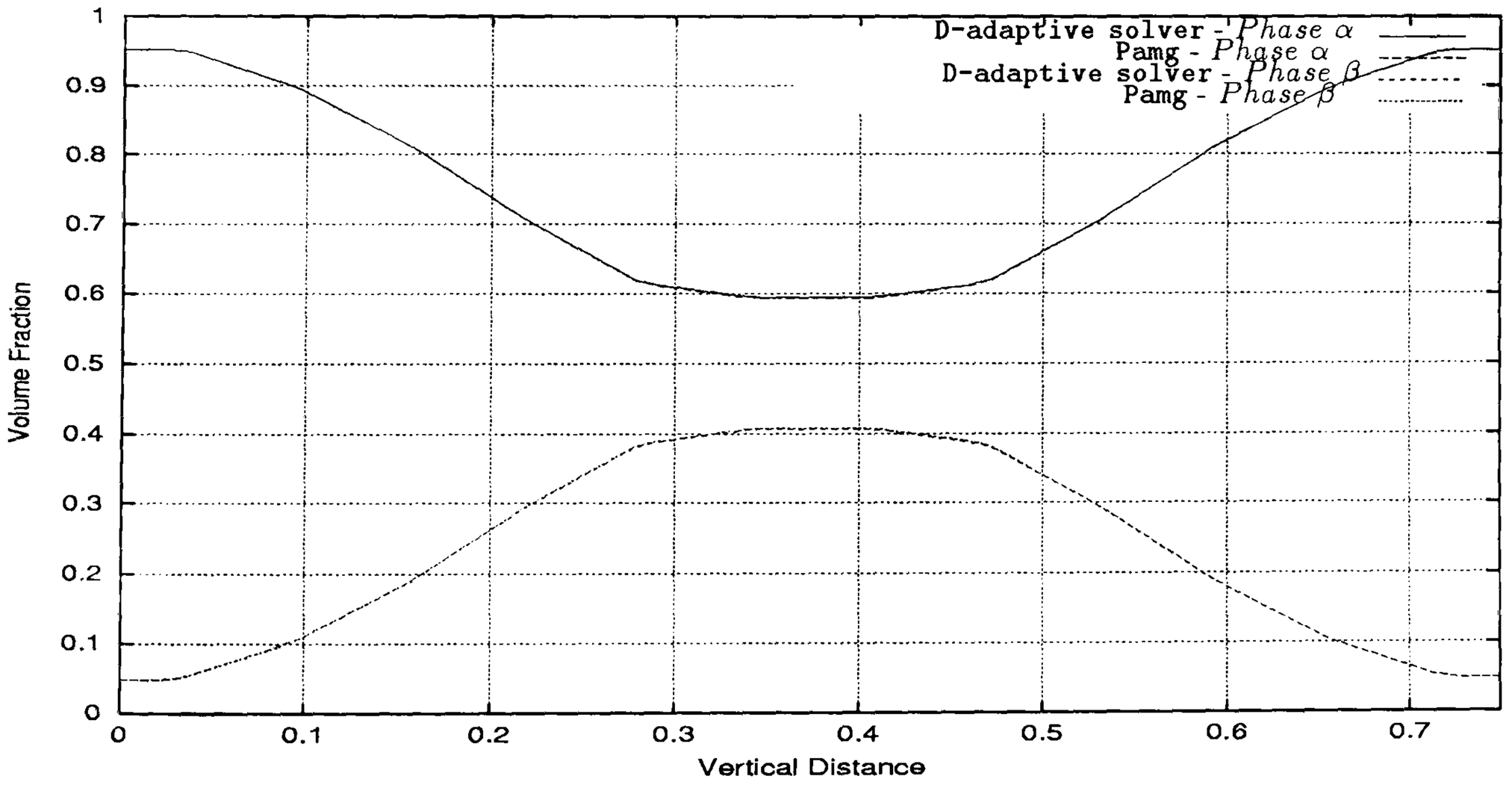


Figure 6.52: Two-phase flow through a Contraction (Case II) - comparison between the volume fraction profile along the line $x = 9.350$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be observed the excellent agreement between the results produced by both solvers.

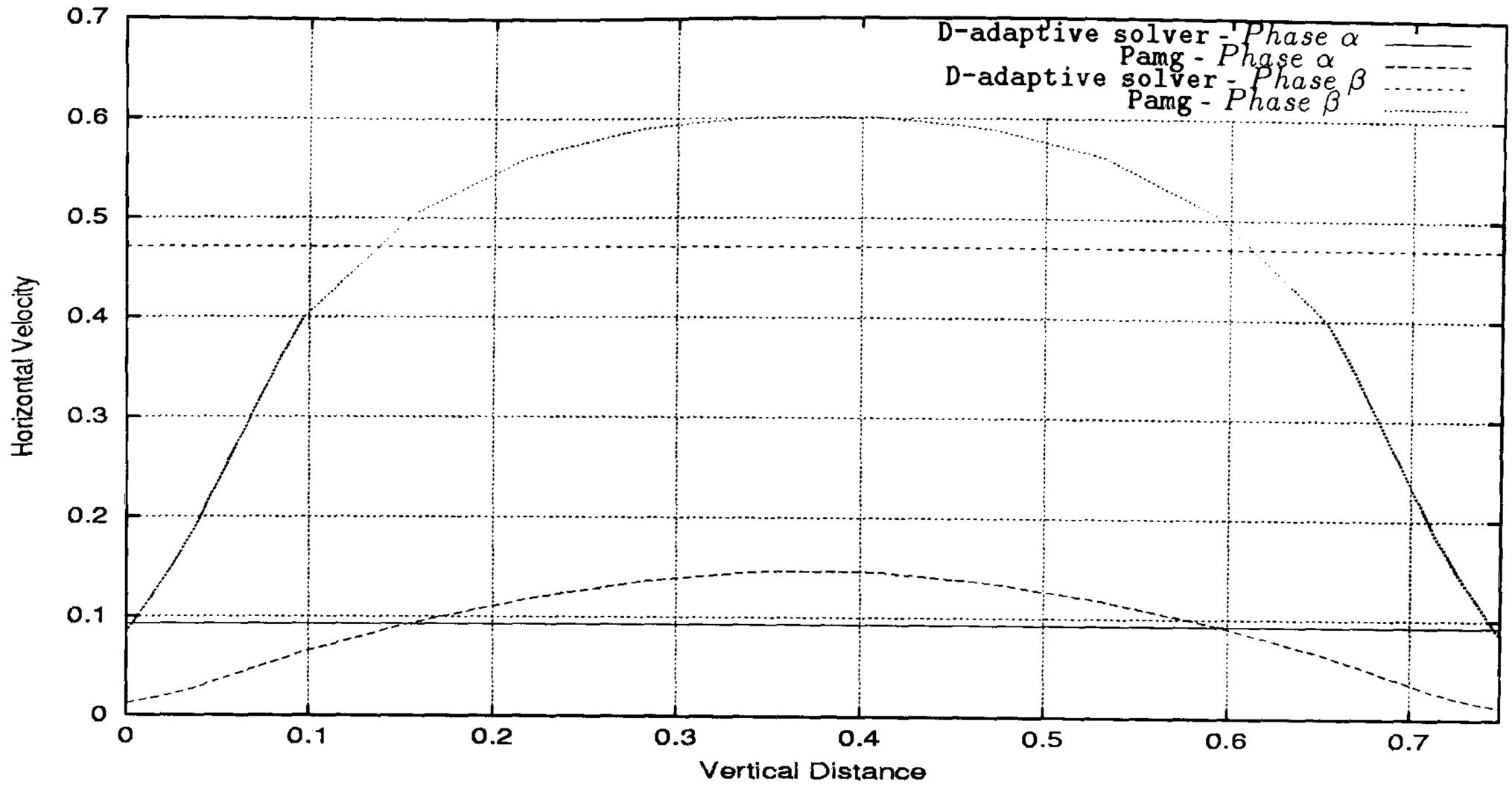


Figure 6.53: Two-phase flow through a Contraction (Case II) - comparison between the horizontal velocity profile along the line $x = 15.350$ for `D-adaptive solver` and `transient Pamg Multi-phase`. It can be seen the parabolic velocity profiles produced by `transient Pamg Multi-phase`. The straight lines produced by the `D-adaptive solver` mean that in this one-dimensional region we only have a value and the concept of vertical distance is not applicable.

and volume fractions at the outlet. The straight lines corresponding to the horizontal velocities and volume fractions produced by the `D-adaptive solver` mean that naturally in the one-dimensional region we only have a value and the concept of vertical distance is not applicable.

Table 6.2 presents the CPU-time spent in the simulation of this test case by the `D-adaptive solver` and the `transient Pamg Multi-phase`. As observed in Table 6.1 the `D-adaptive solver` spents less than 10% of the CPU-time spent by the `transient Pamg Multi-phase` for this test case.

Two-Phase Flow	`D-adaptive solver`	`Pamg`
First One-dimensional domain	231 sec	
Two-dimensional domain	773 sec	
Second One-dimensional domain	232 sec	
Overall domain	1236 sec	22688 sec

Table 6.2: CPU-time spent by the `D-adaptive solver` and the `transient Pamg Multi-phase` in the simulation of the contraction two-phase flow (Case II)

In this section we presented our first version of the `D-adaptive solver`. It has been constructed by coupling the one-dimensional solver `Emaps` with the two-

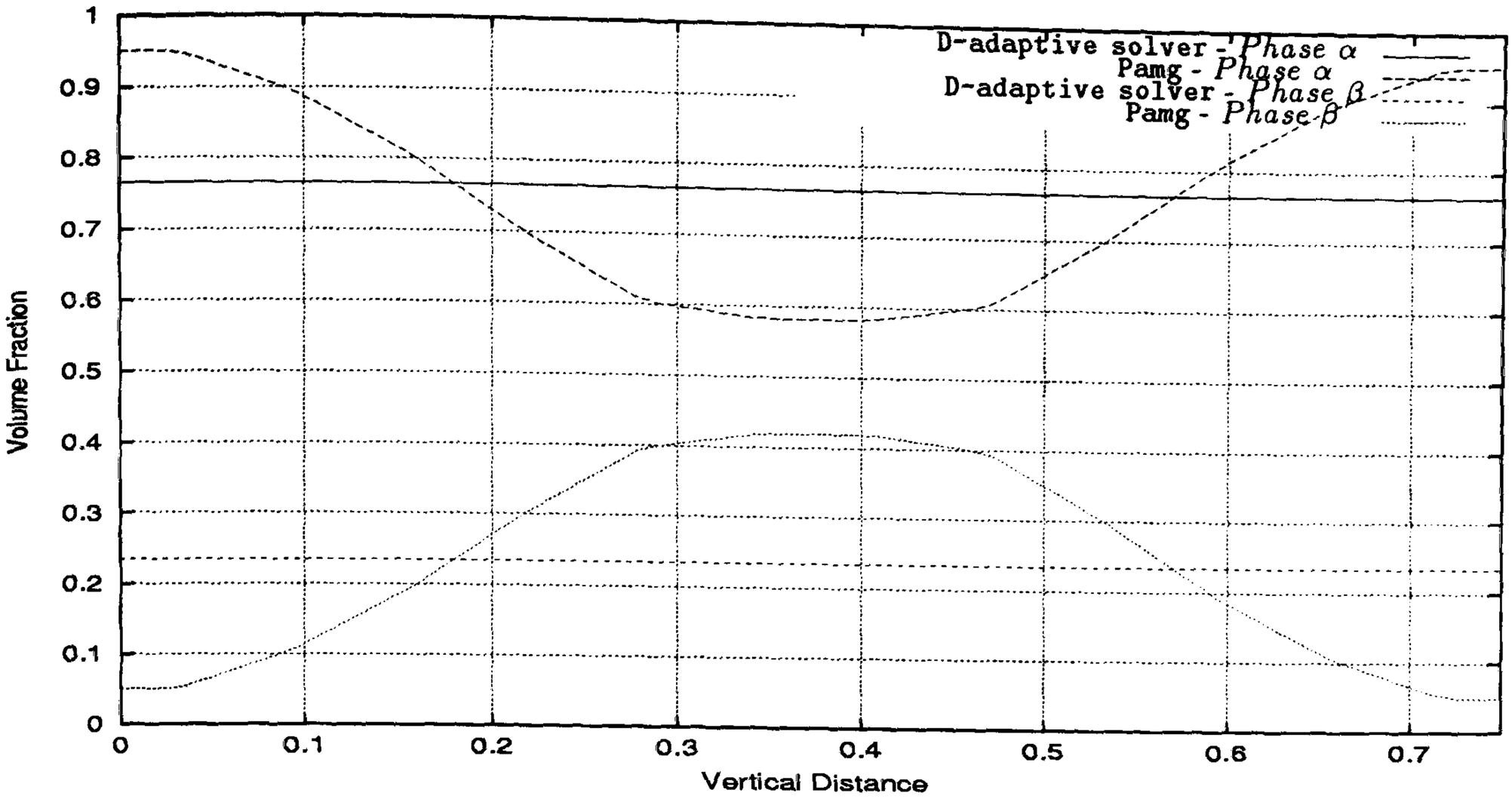


Figure 6.54: Two-phase flow through a Contraction (Case II) - comparison between the volume fraction profile along the line $x = 15.350$ for `D-adaptive solver` and `transient Pamg Multi-phase`. The straight lines produced by the `D-adaptive solver` mean that in this one-dimensional region we only have a value and the concept of vertical distance is not applicable.

dimensional solver `transient Pamg Multi-phase`. The validation of this `D-adaptive solver` has been established by comparing its results with the results produced by the `transient Pamg Multi-phase` for the same test problem. The geometry of this problem has been a contraction, which has proven to be very suitable to apply the `D-adaptive solver`. The existence of regions in the domain where the fluid flow exhibits velocities with the vertical component null suggests that. Hence, observing the solutions of the simulation of a two-phase fluid flow through the contraction produced by `transient Pamg Multi-phase` it has been possible to establish the location of the dimensional interfaces. In these interfaces the formulas derived and validated in the preceding sections have been applied. Two test cases with fluid flows with different viscosities and densities have been produced. The main advantage of this methodology has been verified by measuring and comparing the CPU-times spent by both solvers for these problems. It has been verified that the `D-adaptive solver` spends less than 10% of the CPU-time spent by `transient Pamg Multi-phase`. This has been concluded for both test cases. In the author's opinion additional gains in the performance of `D-adaptive solver` could be achieved if the domain of the problem is bigger than the domain of our test problem. However some important differences have been verified in the results produced by both solvers. One is on the pressure drop issue. Evidence suggests that in the contraction region the pressure drop produced by both solvers are similar. However in the one-dimensional regions the straight lines corresponding to both solvers are not parallel. While `transient Pamg Multi-phase`

produces a relevant pressure drop the **D-adaptive solver** does not. The reason for this in the author's opinion is due to the fact that **Emaps** is prepared for working with pipes and consequently with cylindrical coordinates while **transient Pamg Multi-phase** is prepared for working with Cartesian coordinates. Another difference that has been observed between the results produced by both solvers is the little jumps of the curves produced by the **D-adaptive solver** when changing from the two-dimensional domain to the one-dimensional one. This must be due to the interpolation errors produced in this interface.

So far we have produced and validated in a very complex test problem the first version of our **D-adaptive solver**. Good gains in the performance have been verified. However the two coordinate systems, Cartesian and cylindrical, used by this version of the **D-adaptive solver** may be the main reason for the differences observed. Hence, the decision of constructing the second version of the **D-adaptive solver** which will work entirely in cylindrical coordinates. Since **CFX 4.3** is a solver prepared to work in cylindrical coordinates, we decided to replace the **transient Pamg Multi-phase** solver in the **D-adaptive solver** by **CFX 4.3**. Naturally we need to derive the formulas to use in the one-dimensional/two-dimensional interface in cylindrical coordinates. This is precisely the objective of section 6.3.4.

6.3.4 Steady well-mixed two-phase flow in a cylinder

In this Section we are going to derive in cylindrical coordinates the asymptotic analysis of well-mixed two-phase flow which we earlier performed in cartesian coordinates (see Section 6.3.1). Naturally the main objective is to achieve the correct formulas to apply when changing from a one-dimensional domain to a two-dimensional one when simulating multi-phase fluid flows through a pipe with a contraction. The methodology used to deal with the cylindrical geometry and the two-dimensional treatment was to consider an axially symmetric problem. The symmetry axis is the z direction and consequently the fluid flow moves through that direction. Hence the velocity of the fluid is going to depend on the value on the z and on the distance to that axis the radius distance. Two velocity components can be thought of in this system - the radial velocity and the axial velocity.

Naturally the starting point for this approach is, as in the cartesian version, the set of multi-fluid differential equations derived in Section 3.3. Hence the governing equations for steady well-mixed, uni-directional flow are[80] :

$$\mu_\alpha \frac{1}{r}\frac{\partial}{\partial r}\left(r s_\alpha \frac{\partial w_\alpha}{\partial r}\right) = s_\alpha \frac{\partial p}{\partial z} - \frac{C_D}{d_{\alpha\beta}} s_\alpha s_\beta (s_\alpha \rho_\alpha + s_\beta \rho_\beta)(w_\beta - w_\alpha)|w_\beta - w_\alpha| \quad (6.72)$$

$$\mu_\beta \frac{1}{r}\frac{\partial}{\partial r}\left(r s_\beta \frac{\partial w_\beta}{\partial r}\right) = s_\beta \frac{\partial p}{\partial z} + \frac{C_D}{d_{\alpha\beta}} s_\alpha s_\beta (s_\alpha \rho_\alpha + s_\beta \rho_\beta)(w_\beta - w_\alpha)|w_\beta - w_\alpha| \; . \quad (6.73)$$

where s_α and s_β represent the volume fractions for phases α and β and w_α and w_β the axial component of the velocities for phases α and β. The governing equations for flow in the r direction indicates $p = p(z)$. The system is closed with the following relations

$$s_\alpha + s_\beta = 1 \quad (6.74)$$

$$Q_\alpha = 2\pi \int_0^R s_\alpha w_\alpha \; r \; dr \quad (6.75)$$

$$Q_\beta = 2\pi \int_0^R s_\beta w_\beta \; r \; dr \; . \quad (6.76)$$

where R is the radius of the domain.

Small drag expansion

Since the flow is unidirectional there can be no vertical mixing, hence s_α and s_β must remain constant. Let $C_D/d_{\alpha\beta} = \lambda \ll 1$ and look for solutions of the form

$$w = w_0(r) + \lambda w_1(r) + \cdots \quad (6.77)$$

$$s = s_0 + \lambda s_1(r) + \cdots \quad (6.78)$$

$$p = p_0(z) + \lambda p_1(z) + \cdots \; . \quad (6.79)$$

The leading order problem is governed by

$$\mu_\alpha \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial w_{\alpha 0}}{\partial r}\right) = \frac{\partial p_0}{\partial z} \tag{6.80}$$

$$\mu_\beta \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial w_{\beta 0}}{\partial r}\right) = \frac{\partial p_0}{\partial z} \tag{6.81}$$

$$s_{\alpha 0} + s_{\beta 0} = 1 \tag{6.82}$$

$$Q_\alpha = 2\pi \int_0^R s_\alpha w_{\alpha 0}\, r\, dr \tag{6.83}$$

$$Q_\beta = 2\pi \int_0^R s_\beta w_{\beta 0}\, r\, dr\ . \tag{6.84}$$

Integrating the first two equations gives the standard result

$$w_{\alpha 0} = \frac{1}{4\mu_\alpha}\frac{\partial p_0}{\partial z}(r^2 - R^2) \tag{6.85}$$

$$w_{\beta 0} = \frac{1}{4\mu_\beta}\frac{\partial p_0}{\partial z}(r^2 - R^2)\ . \tag{6.86}$$

Integrating the flux equations and then integrating with respect to z gives an expression for the pressure

$$p_0 = p_a - \frac{8\mu_\alpha Q_\alpha}{\pi s_{\alpha 0} R^4} z = p_a - \frac{8\mu_\beta Q_\beta}{\pi s_{\beta 0} R^4} z\ . \tag{6.87}$$

Hence $s_{\alpha 0}\mu_\beta Q_\beta = s_{\beta 0}\mu_\alpha Q_\alpha$. Substituting this into (6.74) gives

$$s_{\alpha 0} = \frac{\mu_\alpha Q_\alpha}{\mu_\alpha Q_\alpha + \mu_\beta Q_\beta} \qquad s_{\beta 0} = \frac{\mu_\beta Q_\beta}{\mu_\alpha Q_\alpha + \mu_\beta Q_\beta}\ . \tag{6.88}$$

The leading order problem is now completely solved.

The first order equation is governed by

$$\frac{\mu_\alpha}{r}\frac{\partial}{\partial r}\left[r s_{\alpha 1}\frac{\partial w_{\alpha 0}}{\partial r} + r s_{\alpha 0}\frac{\partial w_{\alpha 1}}{\partial r}\right] = s_{\alpha 1}\frac{\partial p_0}{\partial z} + s_{\alpha 0}\frac{\partial p_1}{\partial z} - g_0 \tag{6.89}$$

$$\frac{\mu_\beta}{r}\frac{\partial}{\partial r}\left[r s_{\beta 1}\frac{\partial w_{\beta 0}}{\partial r} + r s_{\beta 0}\frac{\partial w_{\beta 1}}{\partial r}\right] = s_{\beta 1}\frac{\partial p_0}{\partial z} + s_{\beta 0}\frac{\partial p_1}{\partial z} + g_0 \tag{6.90}$$

$$s_{\alpha 1} + s_{\beta 1} = 0 \tag{6.91}$$

$$\int_0^R (s_{\alpha 1} w_{\alpha 0} + s_{\alpha 0} w_{\alpha 1}) r\, dr = \int_0^R (s_{\beta 1} w_{\beta 0} + s_{\beta 0} w_{\beta 1}) r\, dr = 0\ , \tag{6.92}$$

where g_0 is

$$g_0 = s_{\alpha 0} s_{\beta 0}(\rho_\alpha s_{\alpha 0} + \rho_\beta s_{\beta 0})(w_{\beta 0} - w_{\alpha 0})|w_{\beta 0} - w_{\alpha 0}| = A'(w_{\beta 0} - w_{\alpha 0})^2\ .$$

In writing this down it is assumed that $w_{\beta 0} > w_{\alpha 0}$. Whilst not losing generality care must be taken that the less viscous fluid is labelled β, *i.e.* we require $\mu_\alpha > \mu_\beta$.

Since $\mu_\alpha \frac{\partial w_{\alpha 0}}{\partial r} = \mu_\beta \frac{\partial w_{\beta 0}}{\partial r} = r\frac{\partial p_0}{\partial z}/2$, equations (6.89) and (6.90) reduce to

$$\frac{1}{r}\frac{\partial}{\partial r}\left[r\mu_\alpha s_{\alpha 0}\frac{\partial w_{\alpha 1}}{\partial r}\right] = s_{\alpha 0}\frac{\partial p_1}{\partial z} - g_0 \ , \tag{6.93}$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left[r\mu_\beta s_{\beta 0}\frac{\partial w_{\beta 1}}{\partial r}\right] = s_{\beta 0}\frac{\partial p_1}{\partial z} + g_0 \ . \tag{6.94}$$

The function g_0 depends on the velocities $w_{\beta 0}, w_{\alpha 0}$, given by equations (6.85), (6.86). We may write g_0 in the form

$$g_0 = \mu_\alpha s_{\alpha 0} B_\alpha (r^2 - R^2)^2 = \mu_\beta s_{\beta 0} B_\beta (r^2 - R^2)^2 \ ,$$

where

$$B_\alpha = \frac{s_{\beta 0}(\rho_\alpha s_{\alpha 0} + \rho_\beta s_{\beta 0})}{\mu_\alpha}\frac{1}{16}\left(\frac{\partial p_0}{\partial z}\right)^2\left(\frac{1}{\mu_\beta} - \frac{1}{\mu_\alpha}\right)^2$$

$$B_\beta = \frac{s_{\alpha 0}(\rho_\alpha s_{\alpha 0} + \rho_\beta s_{\beta 0})}{\mu_\beta}\frac{1}{16}\left(\frac{\partial p_0}{\partial z}\right)^2\left(\frac{1}{\mu_\beta} - \frac{1}{\mu_\alpha}\right)^2 = \frac{\mu_\alpha}{\mu_\beta}B_\alpha$$

are constant. The velocity equations become

$$\frac{1}{r}\frac{\partial}{\partial r}\left[r\frac{\partial w_{\alpha 1}}{\partial r}\right] = \frac{1}{\mu_\alpha}\frac{\partial p_1}{\partial z} - B_\alpha (r^2 - R^2)^2 \tag{6.95}$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left[r\frac{\partial w_{\beta 1}}{\partial r}\right] = \frac{1}{\mu_\beta}\frac{\partial p_1}{\partial z} + B_\beta (r^2 - R^2)^2 \ . \tag{6.96}$$

These may be integrated immediately to give the first order velocities

$$w_{\alpha 1} = \frac{1}{4\mu_\alpha}\frac{\partial p_1}{\partial z}(r^2 - R^2) - \frac{B_\alpha}{72}(r^2 - R^2)(2r^4 - 7r^2R^2 + 11R^4) \tag{6.97}$$

$$w_{\beta 1} = \frac{1}{4\mu_\beta}\frac{\partial p_1}{\partial z}(r^2 - R^2) + \frac{B_\beta}{72}(r^2 - R^2)(2r^4 - 7r^2R^2 + 11R^4) \ . \tag{6.98}$$

Substituting for the velocities into the flux equations and integrating leads to

$$s_{\alpha 0}\mu_\alpha B_\alpha R^4 - 2s_{\alpha 1}\frac{\partial p_0}{\partial z} - 2s_{\alpha 0}\frac{\partial p_1}{\partial z} = 0 \tag{6.99}$$

$$s_{\beta 0}\mu_\beta B_\beta R^4 + 2s_{\beta 1}\frac{\partial p_0}{\partial z} + 2s_{\beta 0}\frac{\partial p_1}{\partial z} = 0 \ . \tag{6.100}$$

Eliminating the pressure gradient provides expressions for the volume fractions

$$s_{\alpha 1} = \frac{\mu_\alpha B_\alpha R^4 s_{\alpha 0} s_{\beta 0}}{\frac{\partial p_0}{\partial z}} = -s_{\beta_1} \ .$$

Eliminating the volume fractions provides an expression for the pressure gradient

$$\frac{\partial p_1}{\partial z} = \mu_\alpha B_\alpha R^4\left(\frac{1}{2} - s_{\beta 0}\right) = \mu_\alpha B_\alpha R^4\left(\frac{\mu_\alpha Q_\alpha - \mu_\beta Q_\beta}{2(\mu_\alpha Q_\alpha + \mu_\beta Q_\beta)}\right) \ .$$

In this section we have derived the new cylindrical version of the expressions for the velocities and volume fractions to apply in the interface between a one-dimensional domain and a two-dimensional domain. The methodology used has been quite naturally similar to that one used in the derivation of the Cartesian version. Hence the starting equations have been the governing equations for steady well-mixed uni-directional flow in cylindrical coordinates. The same assumptions used in the Cartesian case have been made. One is that the fluid flow does not depend on the axial distance. In fact we expect that this happen in a very long pipe. Since the flow is uni-directional it is expected that the volume fractions for each phase remain constant. Hence no vertical mixing is expected to happen. Naturally the parameter to use in the expansion must be very small otherwise the approximation of the velocities with the leading order terms will not be accurate.

The formulas derived above can be very useful in our project. Since we want to couple a one-dimensional solver with a two-dimensional one, these formulas give us expressions for computing the velocities and volume fractions in the interface region. Before using them in the validation of the cylindrical coordinate version of the `D-adaptive solver` let us validate them in a very simple problem. This is the aim of next section.

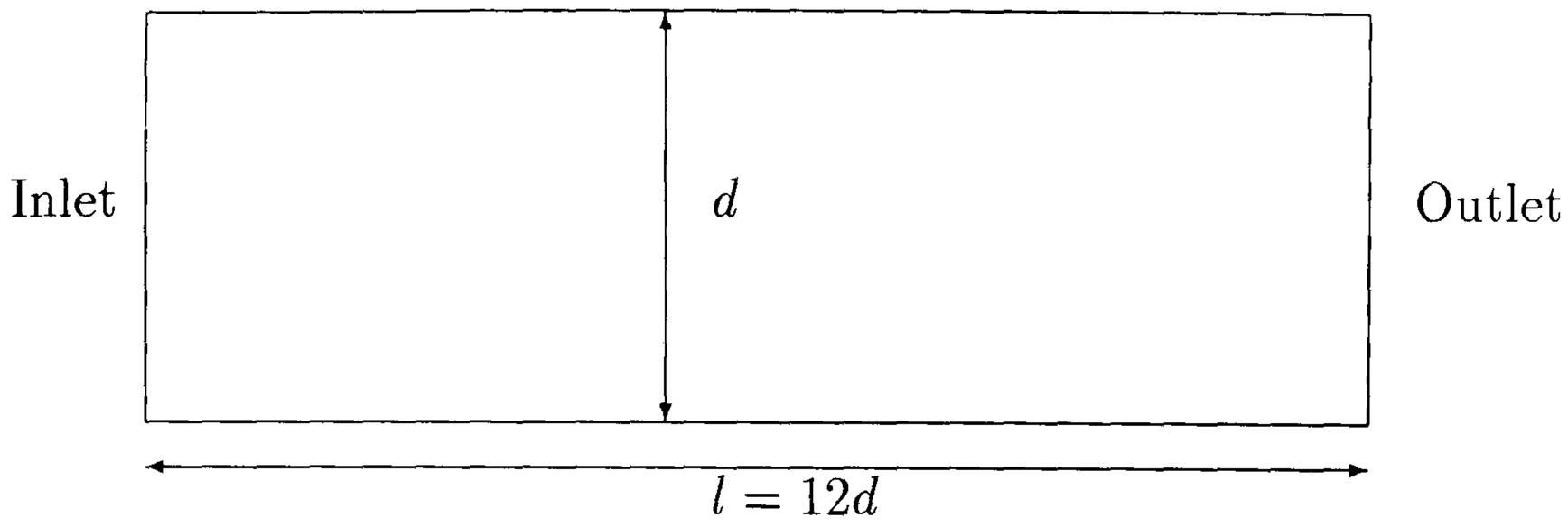


Figure 6.55: Geometrical representation of the domain of the Cylinder two-phase problem

6.3.5 Validation in a Two-Phase Flow through a Cylinder

In this section we are going to validate the asymptotic formulas derived in Section 6.3.4 for the steady two-phase fluid flow in cylindrical coordinates. The methodology used in this validation process is identical to that one used in the validation of the cartesian asymptotic formulas (see Section 6.3.2). Hence, the process relies on the comparison of the computational results with the analytical solution of two-phase fluid flow in a cylinder. A set of fluids with different viscosities has been used for this purpose. The geometry and the boundary conditions used are always the same for each test case.

The geometry for this test case is represented in Figure 6.55. It is simply a straight cylinder of length $12d$ where d is its diameter.

The boundary conditions used are at the inlet, Dirichlet boundary conditions and at the outlet Neumann conditions. In the inlet a parabolic velocity profile is specified for the r-component velocity $u(r, z)$ by

$$u(r, 0) = -7.12(r^2 - (0.75^2))$$

and for the z-component velocity $v(r, z)$ it is null in the boundary, i.e. $v(0, z) = 0$.

The volume fractions in the inlet are set to 0.5 for both phases. In the outlet the z-component velocity $v(r, z)$ is set to zero.

The values chosen for the inter-facial length are $d_{\alpha\beta} = d_{\beta\alpha} = 0.1$. For the drag coefficient four different values have been used: $C_D = 0.0$, $C_D = 0.01$, $C_D = 0.001$ and $C_D = 0.0001$. Hence the corresponding parameter $\lambda = C_D/d_{\alpha\beta}$ satisfy $\lambda \ll 1$ since it takes the values 0.0, 0.1, 0.01 and 0.001.

The physical properties chosen for the first test case are:

Two-Phase Flow	Reynolds Number	Viscosity	Density
Phase α	75	0.01	1.0
Phase β	150	0.005	1.0

Figures 6.56 and 6.57 present the results for the axial velocities and volume fractions produced by `CFX 4.3` and by the analytical expressions when the parameter $\lambda = 0.0$. It can be observed that the volume fractions agree very well over the entire outlet. The velocities exhibit the pattern expected, from 0 to the maximum value when $r = 0.0$ and the agreement between the computational results and the analytically ones is very effective. The relation between the velocities of the two phases in this particular test case is $u_\alpha = 2u_\beta$ which can be observed.

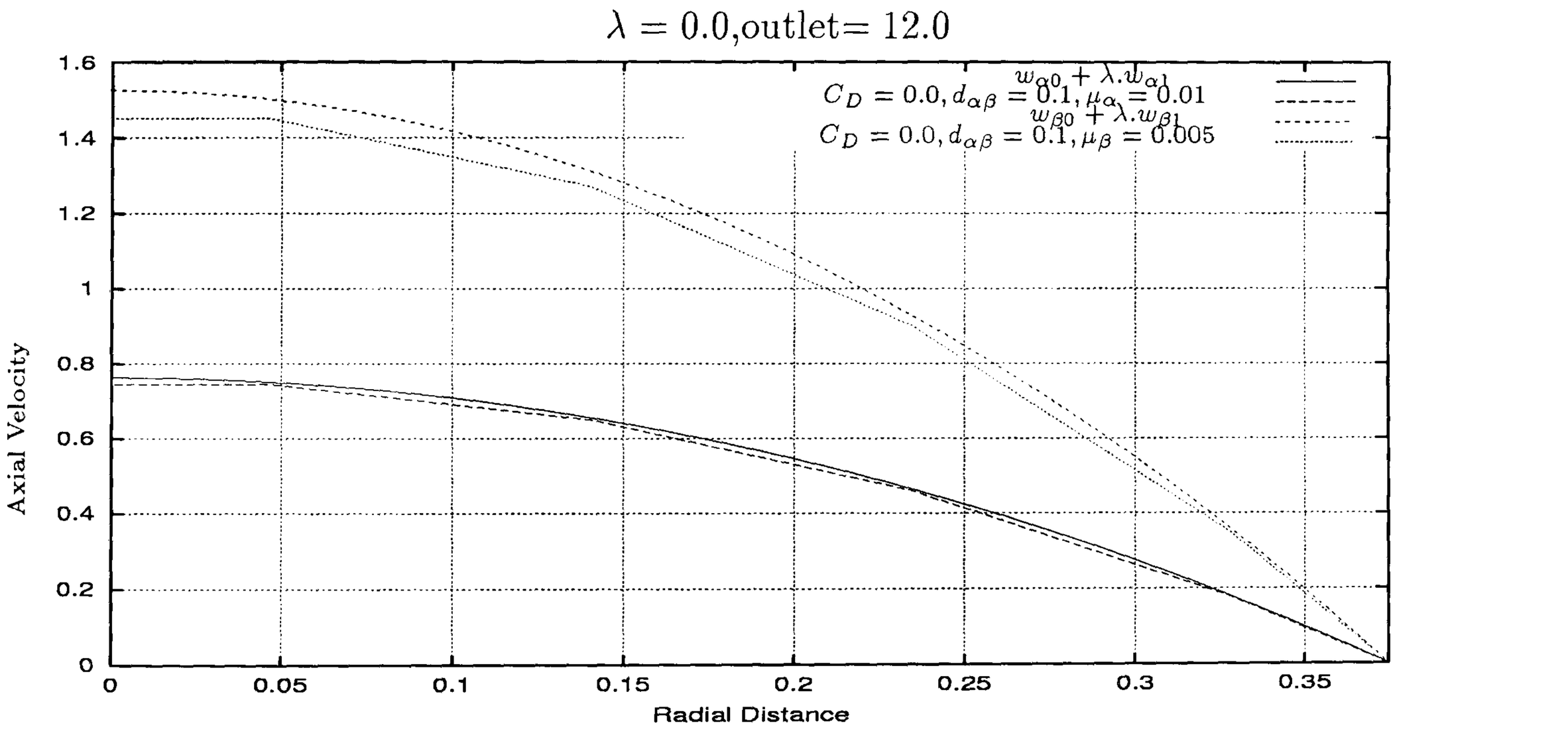


Figure 6.56: Two-phase flow through a Cylinder - comparison between the analytical axial velocity and the computational one along the line $z = 12.0$ for $\lambda = 0.0$. It can be seen good agreement between the profiles correspondent to the analytical and computational velocities for both phases. Note the maximum value when the radial distance is null and the null velocity when the radial distance takes the maximum value.

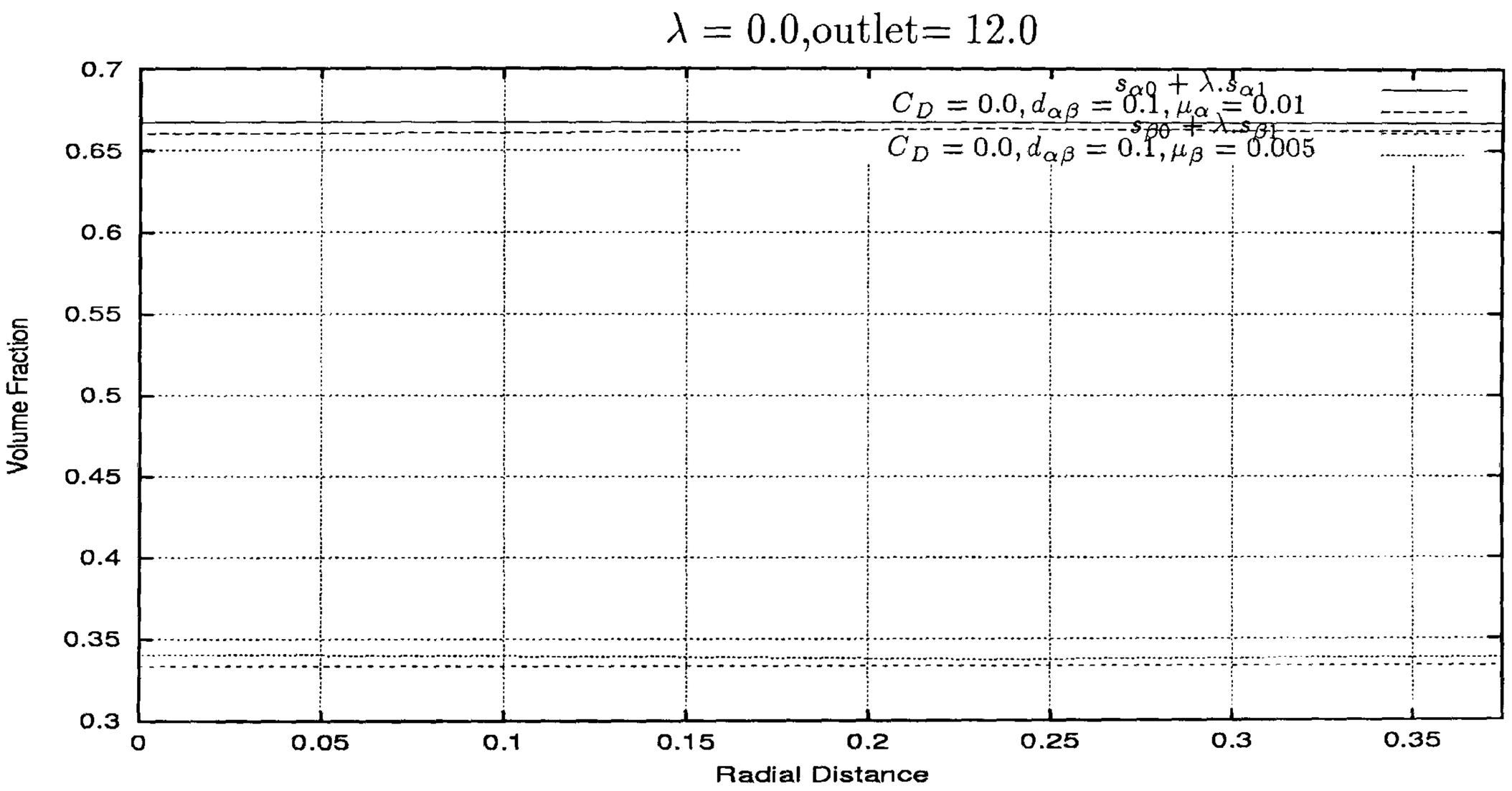


Figure 6.57: Two-phase flow through a Cylinder - comparison between the analytical volume fraction and the computational one along the line $z = 12.0$ for $\lambda = 0.0$. It can be observed that the volume fraction profiles agree very well at this outlet region.

Figures 6.58 through 6.63 present the results for the cases with λ not equal 0.0. Starting from $\lambda = 0.1$ and ending with $\lambda = 0.001$ it can be verified that as the parameter λ becomes smaller, the difference between the computational results and the analytically computed ones diminishes. A particular interesting behaviour can be observed between the radial velocities for the case $\lambda = 0.1$ and $\lambda = 0.01$ (Figures 6.58 and 6.60). In fact the difference between radial velocities of phase α computed by `CFX 4.3` and by the analytical formulas become bigger when changing $\lambda = 0.1$ to $\lambda = 0.01$. However the phase β difference diminuishes. In the author's opinion the reazon for this ocurrence relies in the truncation errors when approximating the radial velocity by the first order velocities.

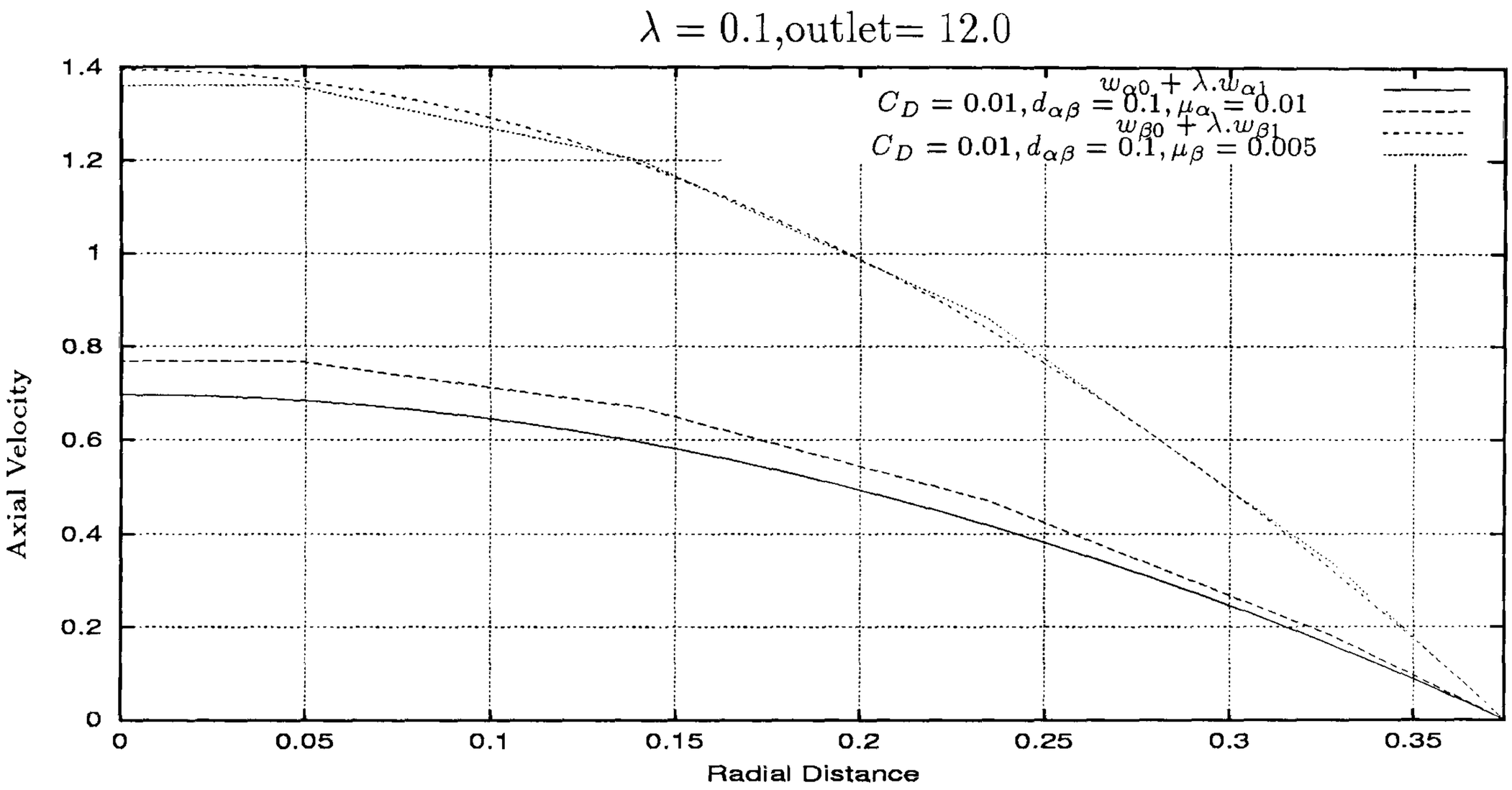


Figure 6.58: Two-phase flow through a Cylinder - comparison between the analytical axial velocity and the computational one along the line $z = 12.0$ for $\lambda = 0.1$. It can be verified the good agreement between the profiles designed for the analytical and computational velocities. For phase α it is excellent.

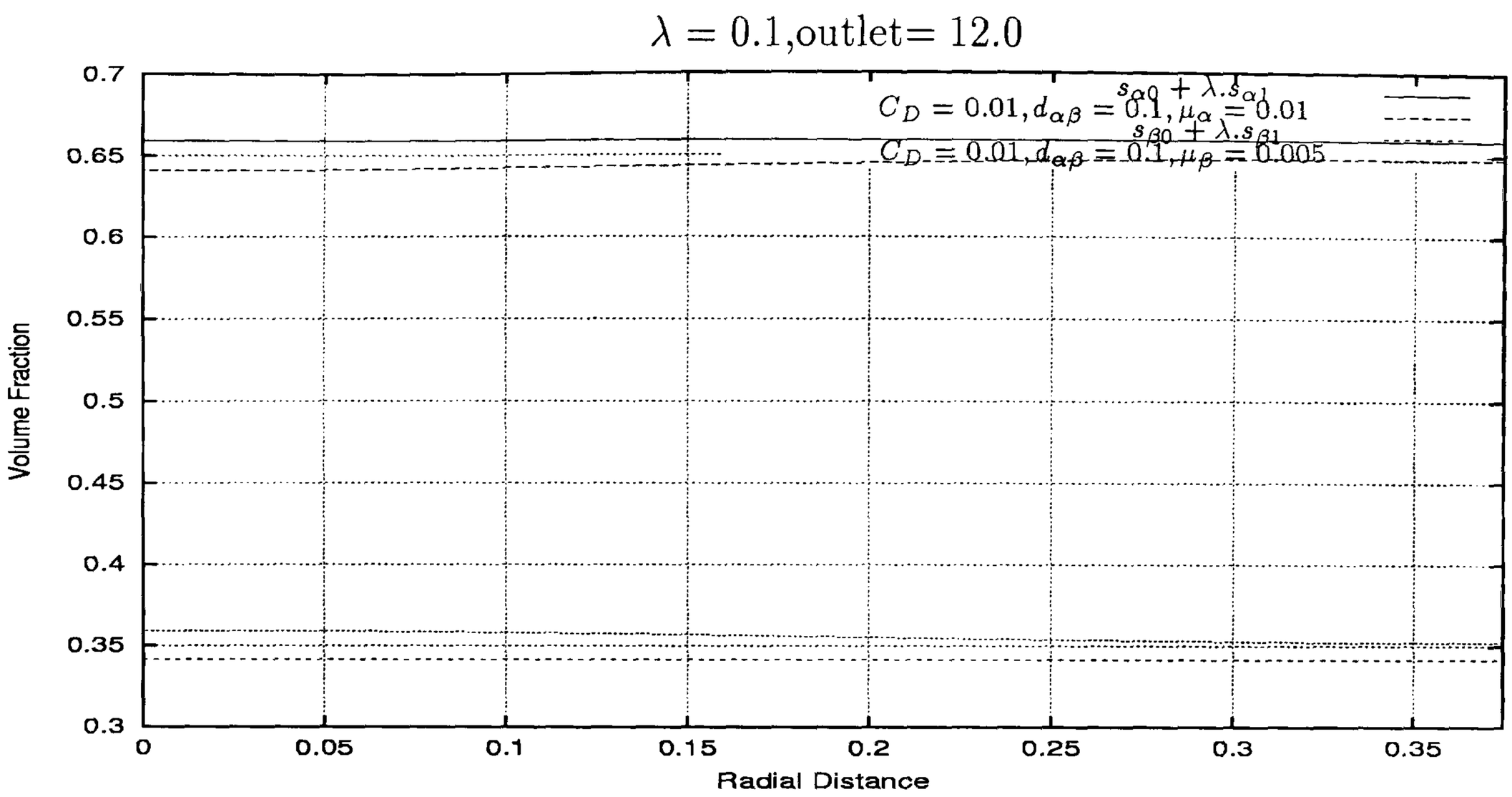


Figure 6.59: Two-phase flow through a Cylinder - comparison between the analytical volume fraction and the computational one along the line $z = 12.0$ for $\lambda = 0.1$. Good agreement can be observed between the profiles produced by the two different ways.

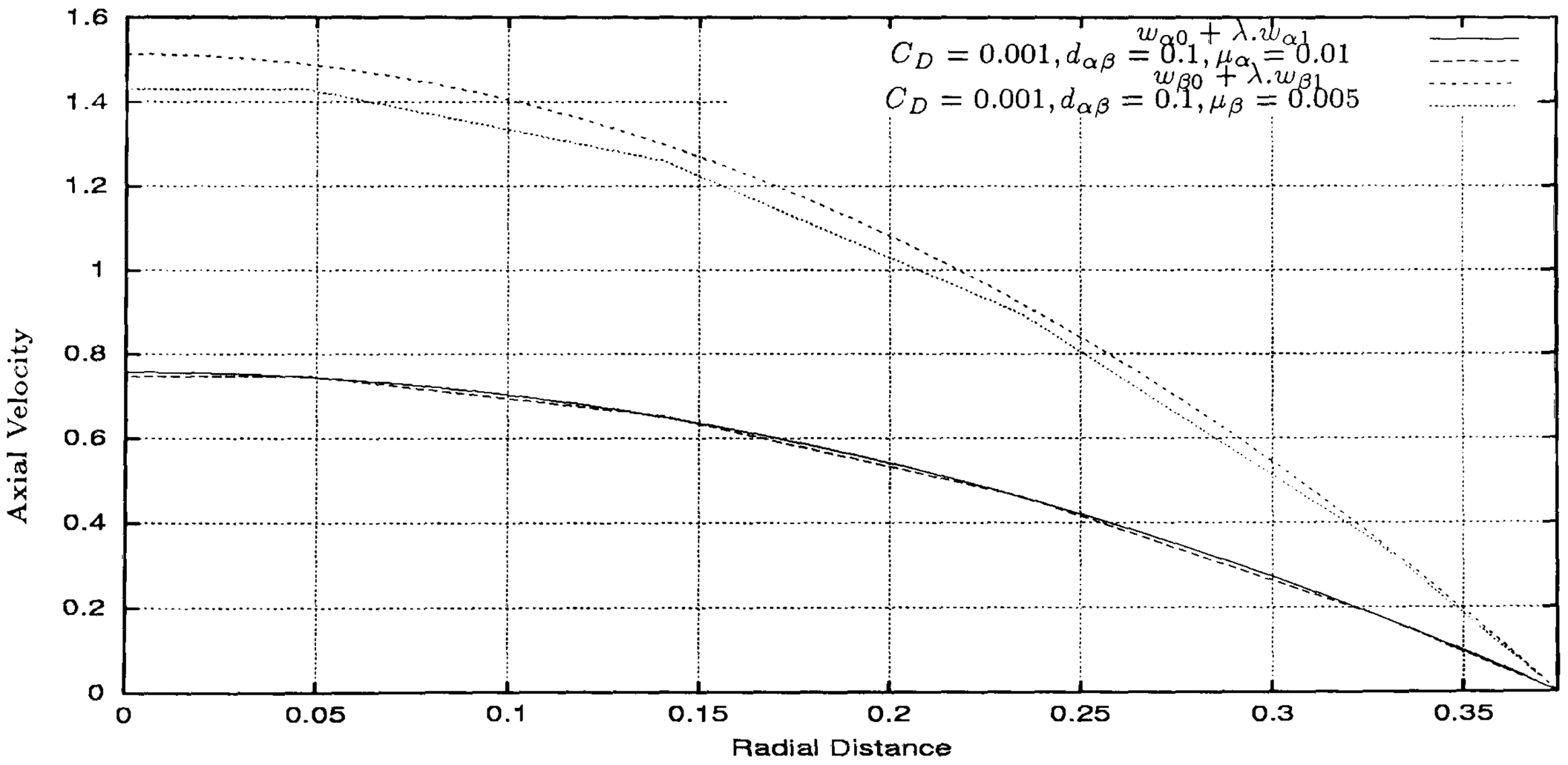


Figure 6.60: Two-phase flow through a Cylinder - comparison between the analytical axial velocity and the computational one along the line $z = 12.0$ for $\lambda = 0.01$. It can be seen good agreement between the analytic and computational axial velocity results. For phase β it is excellent.

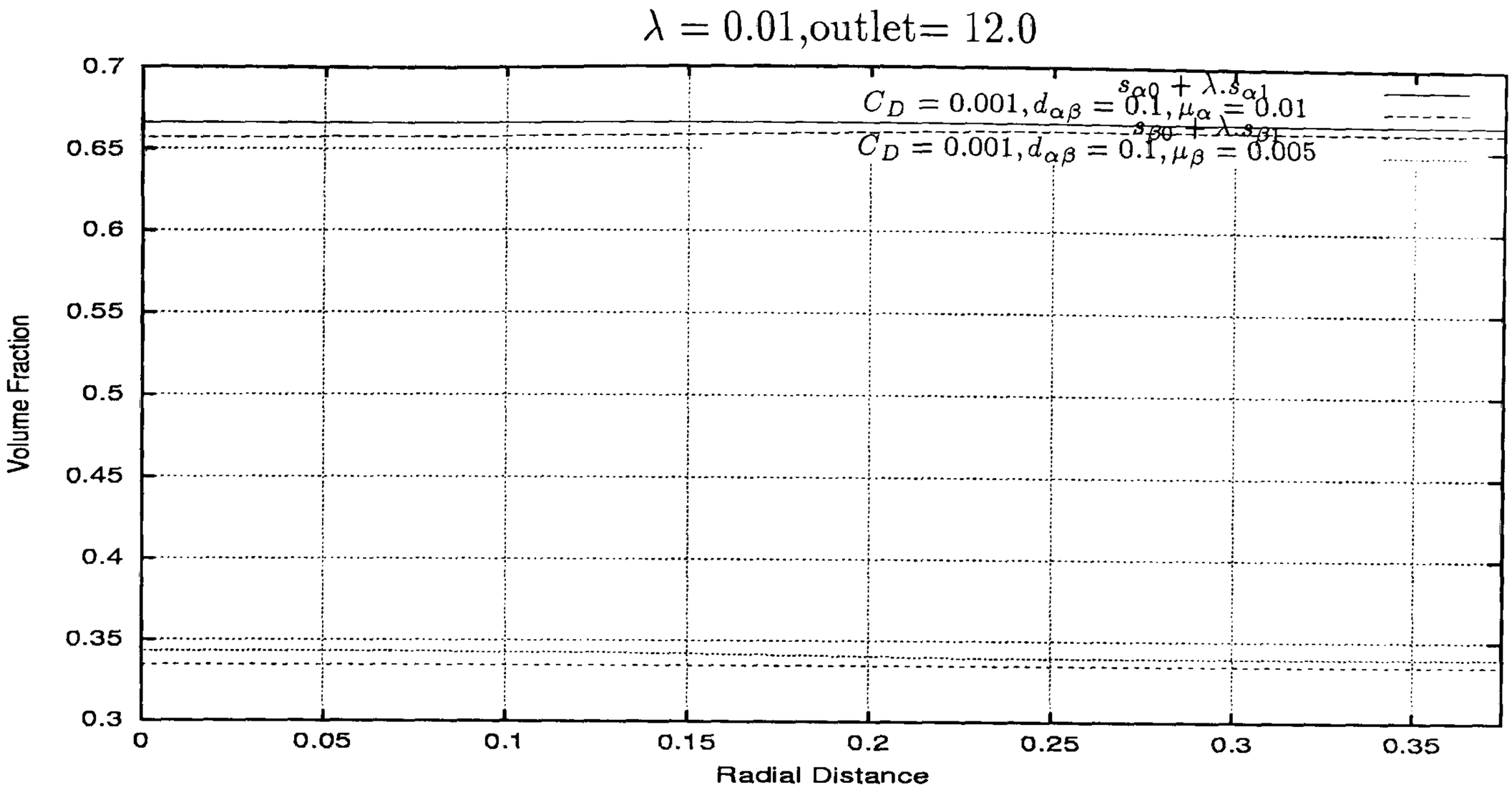


Figure 6.61: Two-phase flow through a Cylinder - comparison between the analytical volume fraction and the computational one along the line $z = 12.0$ for $\lambda = 0.01$. It can be observed good agreement between the volume fractions produced by both manners.

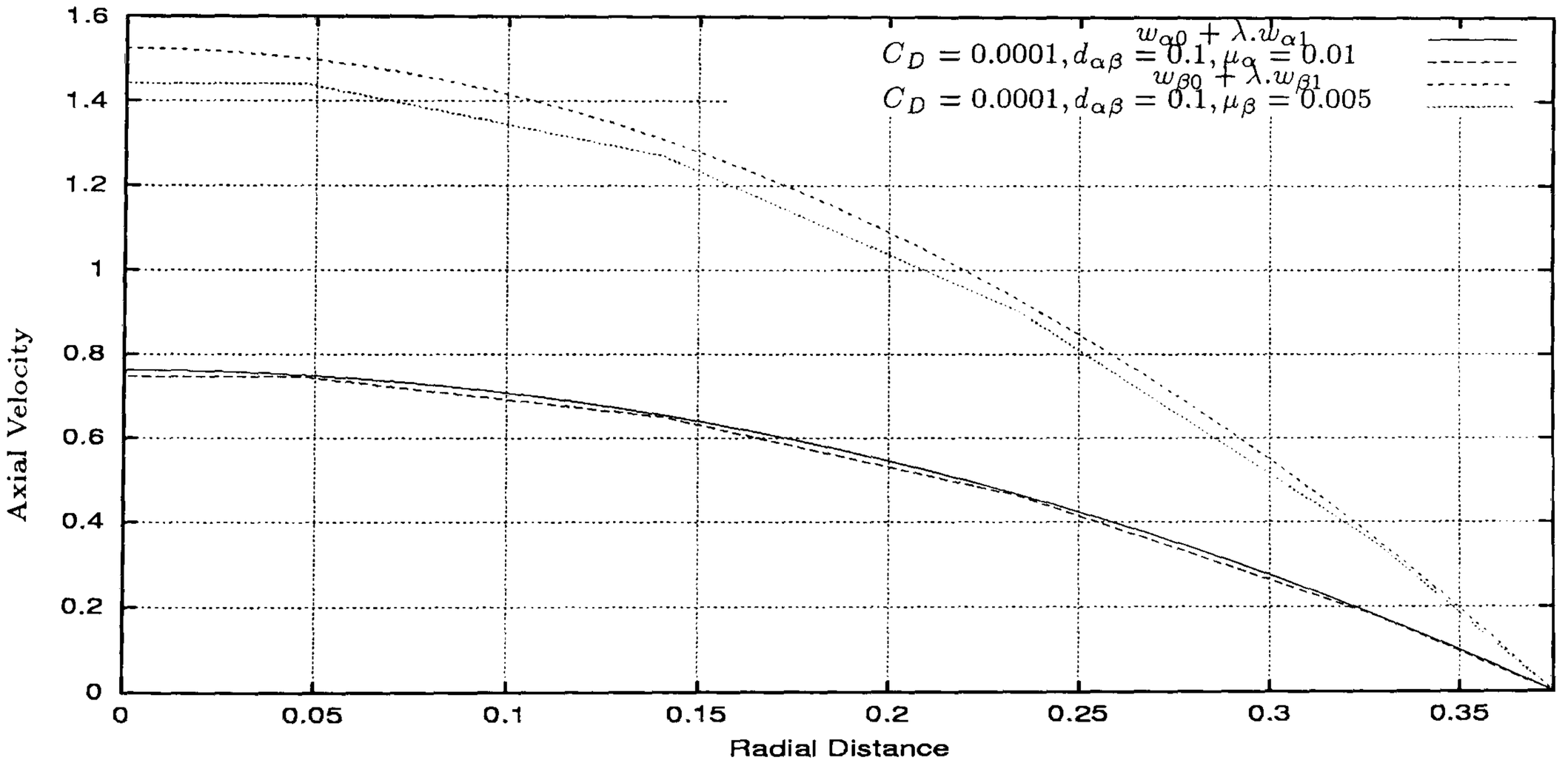


Figure 6.62: Two-phase flow through a Cylinder - comparison between the analytical axial velocity and the computational one along the line $z = 12.0$ for $\lambda = 0.001$. It can be seen good agreement between the analytical and computational axial velocities for both phases.

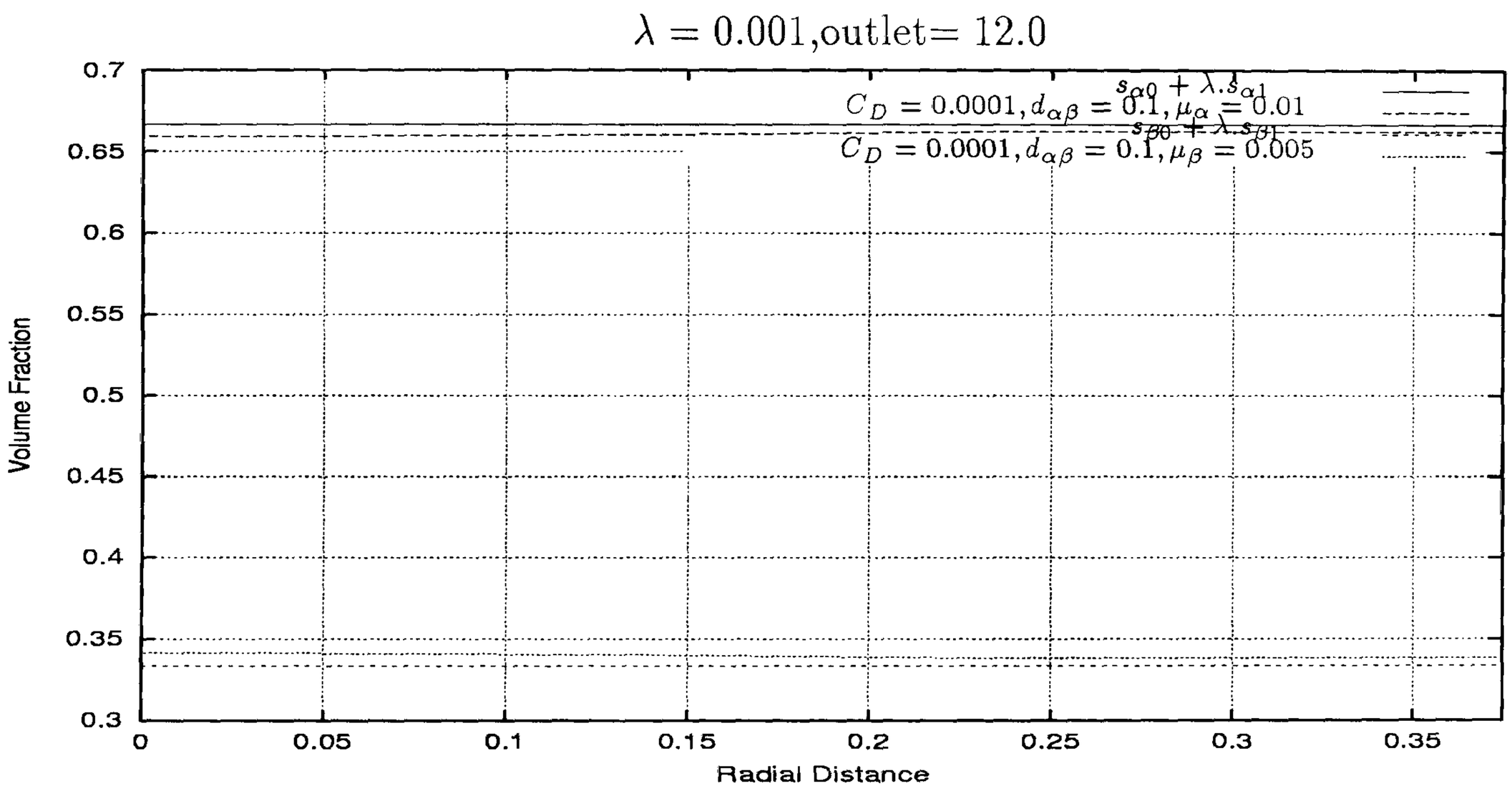


Figure 6.63: Two-phase flow through a Cylinder - comparison between the analytical volume fraction and the computational one along the line $z = 12.0$ for $\lambda = 0.001$. Good agreement is observed between the analytical volume fractions and the computational ones for both phases.

A variation of this test case has been outlined with the purpose of validating the asymptotic formulas for two-phase flows with different densities. Hence the physical properties chosen for this second test case are:

Two-Phase Flow	Reynolds Number	Viscosity	Density
Phase α	75	0.01	1.0
Phase β	300	0.005	0.5

The inter-facial length take the values $d_{\alpha\beta} = d_{\beta\alpha} = 0.1$ and the drag coefficient assumes the same four different values used in the preceding case: $C_D = 0.0$, $C_D = 0.01$, $C_D = 0.001$ and $C_D = 0.0001$. Hence the corresponding parameter λ takes the values 0.0, 0.1, 0.01 and 0.001.

Figures 6.64 and 6.65 presents the comparison between the radial velocities and volume fractions computed by `CFX 4.3` and when $\lambda = 0.0$. The agreement between the radial velocities calculated by the two-different ways is excellent. However the volume fractions produced by `CFX 4.3` have constant values along all the radial distance at the outlet but these constant values are slightly different from the analytically computed ones.

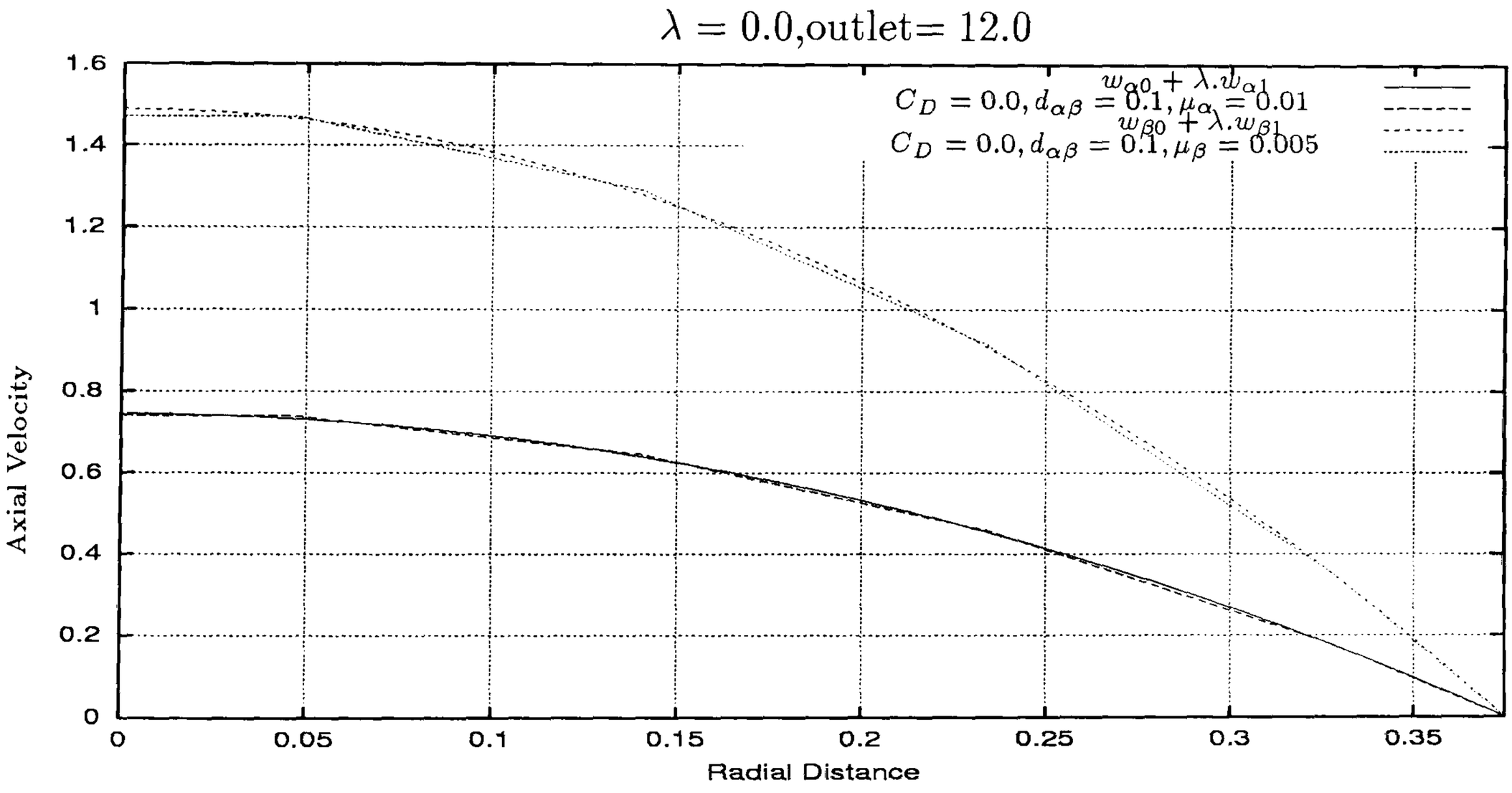


Figure 6.64: Two-phase flow through a Cylinder - comparison between the analytical axial velocity and the computational one along the line $z = 12.0$ for $\lambda = 0.0$. It can be seen excellent agreement between the analytical axial velocities and the computational ones for both phases.

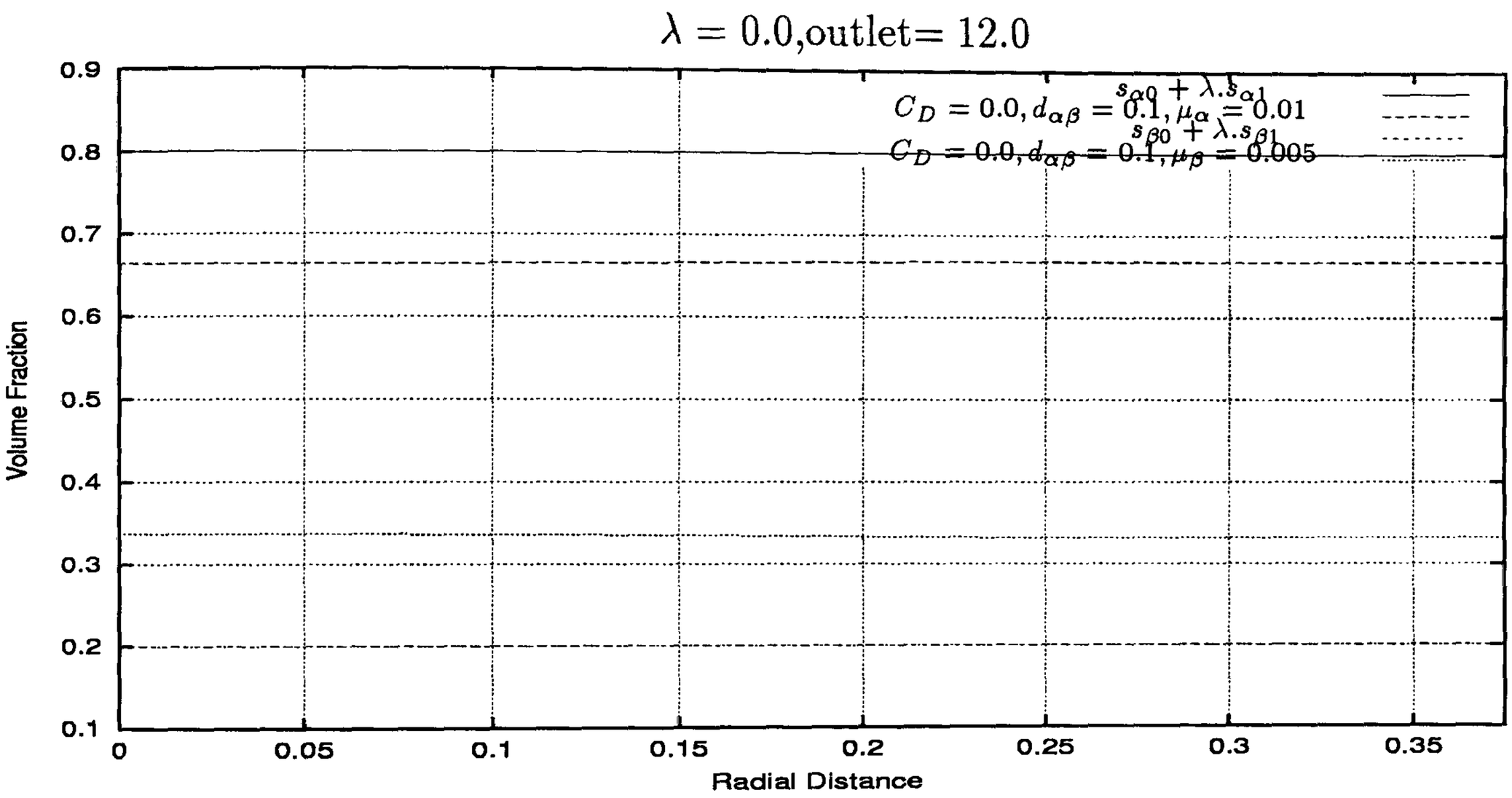


Figure 6.65: Two-phase flow through a Cylinder - comparison between the analytical volume fraction and the computational one along the line $z = 12.0$ for $\lambda = 0.0$. It can be observed that the volume fractions computed analytically and computationally are constant. An important gap between the straight lines correspondent to the analytical and computational calculations can be observed.

Figures 6.66 through 6.71 represent the test cases considered with $\lambda = 0.1$, $\lambda = 0.01$ and $\lambda = 0.001$. The radial velocities profiles become very close when λ reduces its value. Furthermore the radial velocities of the two phases show the same uniform behaviour. This means that when changing from a λ value to a smaller one, the agreement between the curves produced by the solver and the curves produced by the analytic expressions becomes better for both velocities simultaneously. The same behaviour pattern is observed in the volume fraction pictures. However the agreement is not so effective than that observed in the axial velocities.

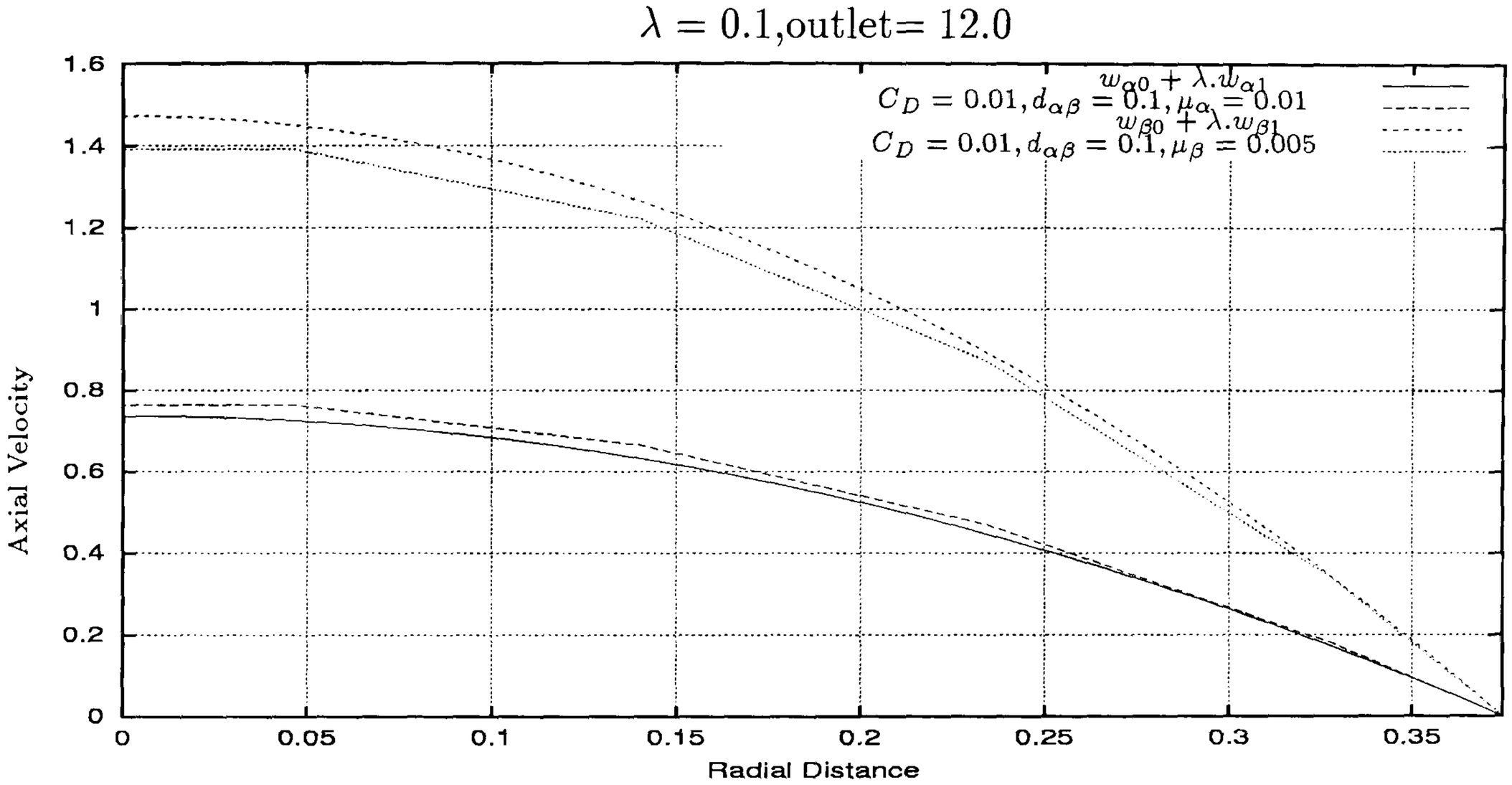


Figure 6.66: Two-phase flow through a Cylinder - comparison between the analytical axial velocity and the computational one along the line $z = 12.0$ for $\lambda = 0.1$. It can be seen good agreement between the analytical axial velocities and the computational ones for both phases.

Finally to finish this validation section of the asymptotic formulas in cylindrical coordinates we made another experiment basically changing the viscosity of one of the fluids of the first test case. Hence the physical properties are:

Two-Phase Flow	Reynolds Number	Viscosity	Density
Phase α	7.5	0.1	1.0
Phase β	75	0.01	1.0

For the drag coefficient the value used has been $C_D = 0.01$ and consequently $\lambda = 0.1$. In Figures 6.72 and 6.73 we present the results for the axial velocity and volume fractions for this case. It is the worst case with respect to the agreement between the curves. This suggests that the fact of having bigger differences in the viscosities of the fluids cause bigger discrepancies between the computational outputs produced by the solver and the analytical results.

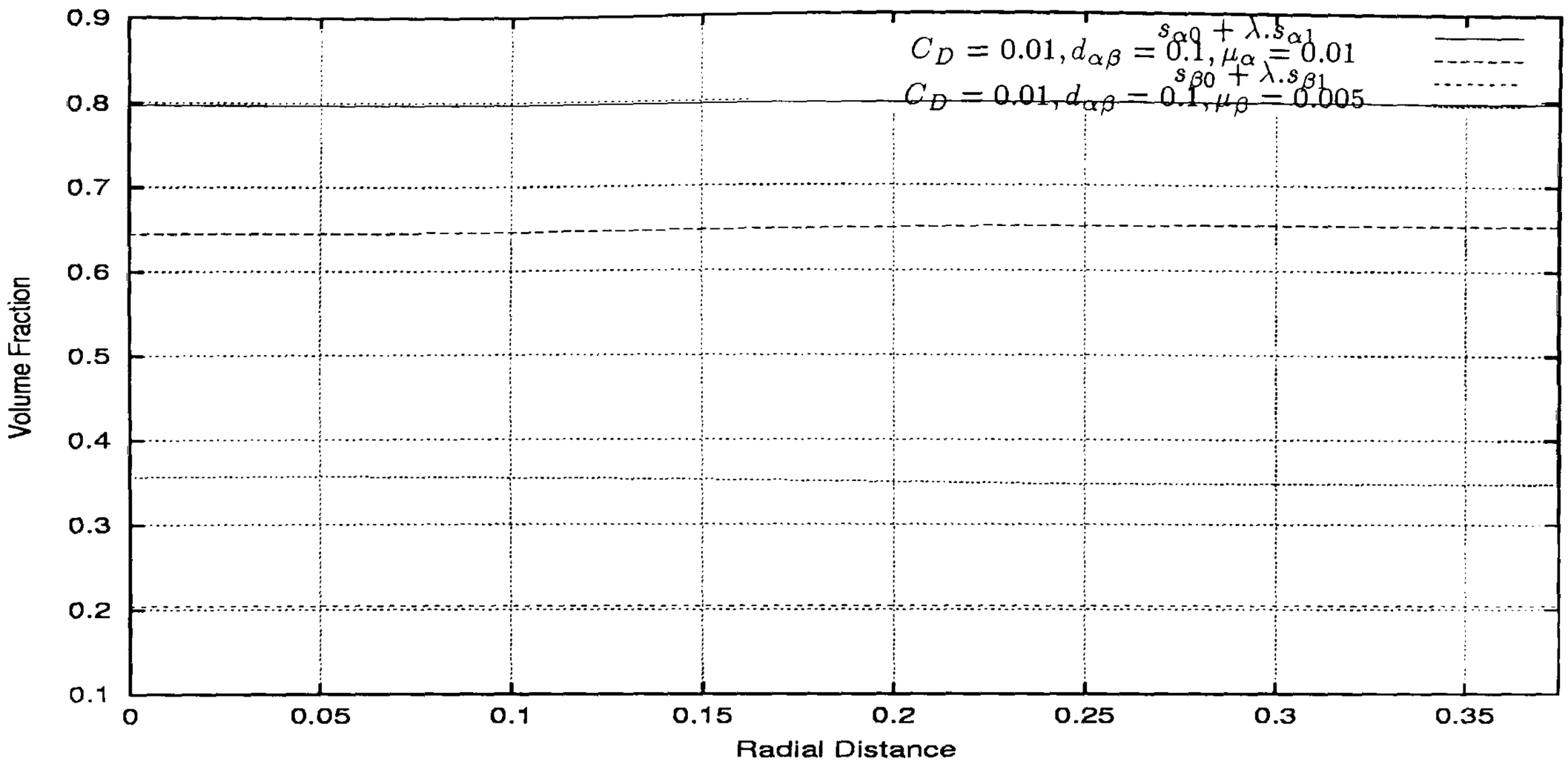


Figure 6.67: Two-phase flow through a Cylinder - comparison between the analytical volume fraction and the computational one along the line $z = 12.0$ for $\lambda = 0.1$. It can be observed that the volume fractions computed by both ways are constant.

In this section we have validated the asymptotic formulas in cylindrical coordinates derived in the previous section. The methodology outlined has been based on the comparison of the results produced analytically and the computational results of a simulation of a two-phase fluid through a cylinder. `CFX 4.3` solver has made the computational simulation in an axially symmetric problem. The geometry of this problem has been very simple. It has been just a straight cylinder. Several sets of fluids with varying viscosities and densities have been used in the test cases. The main conclusions that can be undertaken is that the results produced by the asymptotic formulas agree very well with the computational results produced by `CFX 4.3` solver if the set of fluids have similar values for the densities and viscosities. This agreement is more effective if the parameter used in the expansion, which is related with the drag and inter-facial length, becomes smaller. Strong evidence has been verified of this behaviour. However, it has been verified that for sets of fluids with substantial differences in the densities the volume fractions exhibit substantial differences from the computed values. This must be due to the assumption that there is no vertical mixing. However in this test case the velocities computed by both ways exhibit a good agreement. The last case studied with a two-phase fluid flow with substantially different viscosities showed that substantial differences can be verified between the computational and analytical results.

With this conclusions in mind we are going to implement in a very complex problem the second version of the `D-adaptive solver`, which will work in cylindrical coordinates over all the entire domain of the problems. This is the aim of next section.

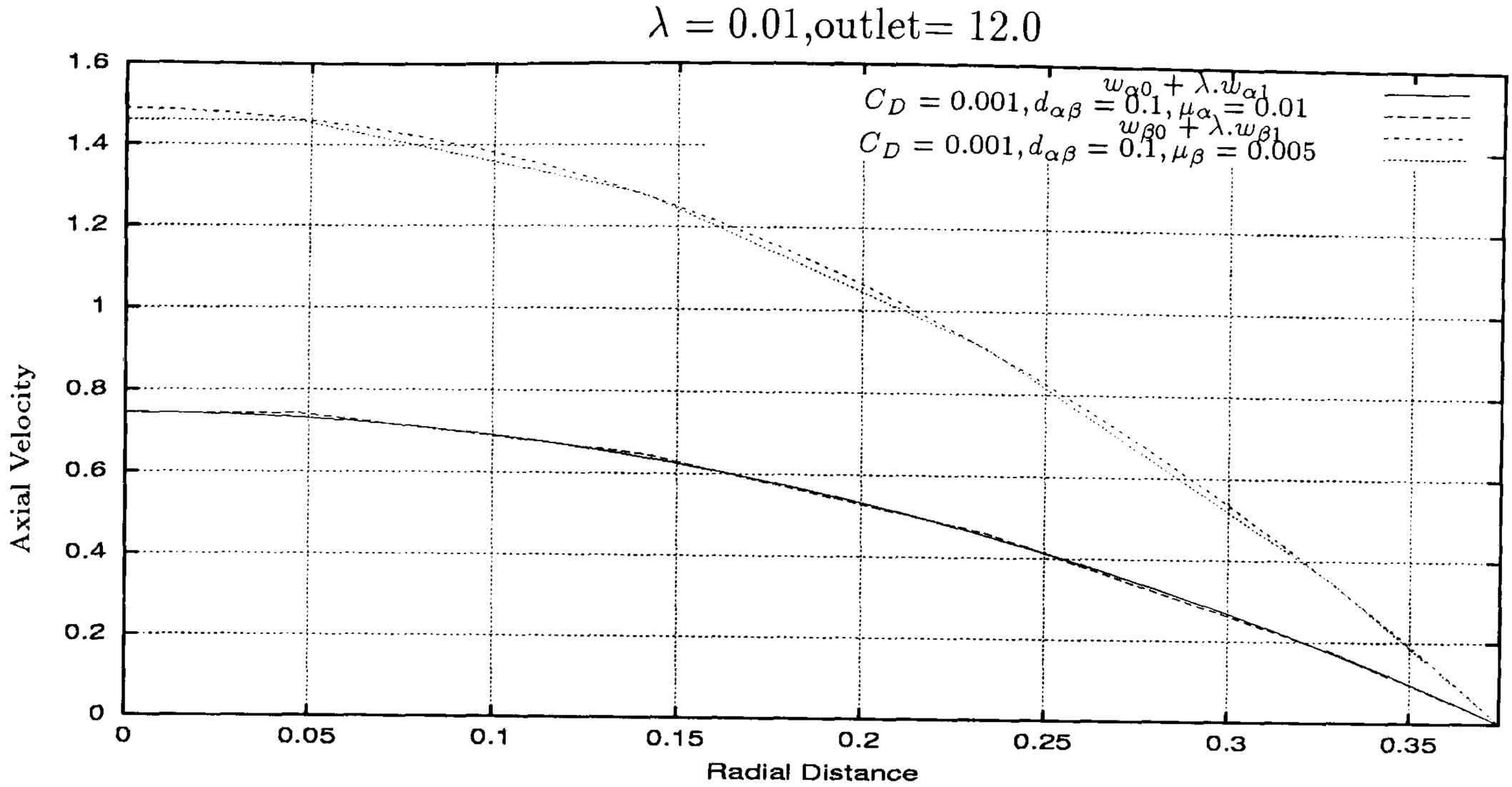


Figure 6.68: Two-phase flow through a Cylinder - comparison between the analytical axial velocity and the computational one along the line $z = 12.0$ for $\lambda = 0.01$. Excellent agreement can be observed between the analytical axial velocities and the computational ones for both phases.

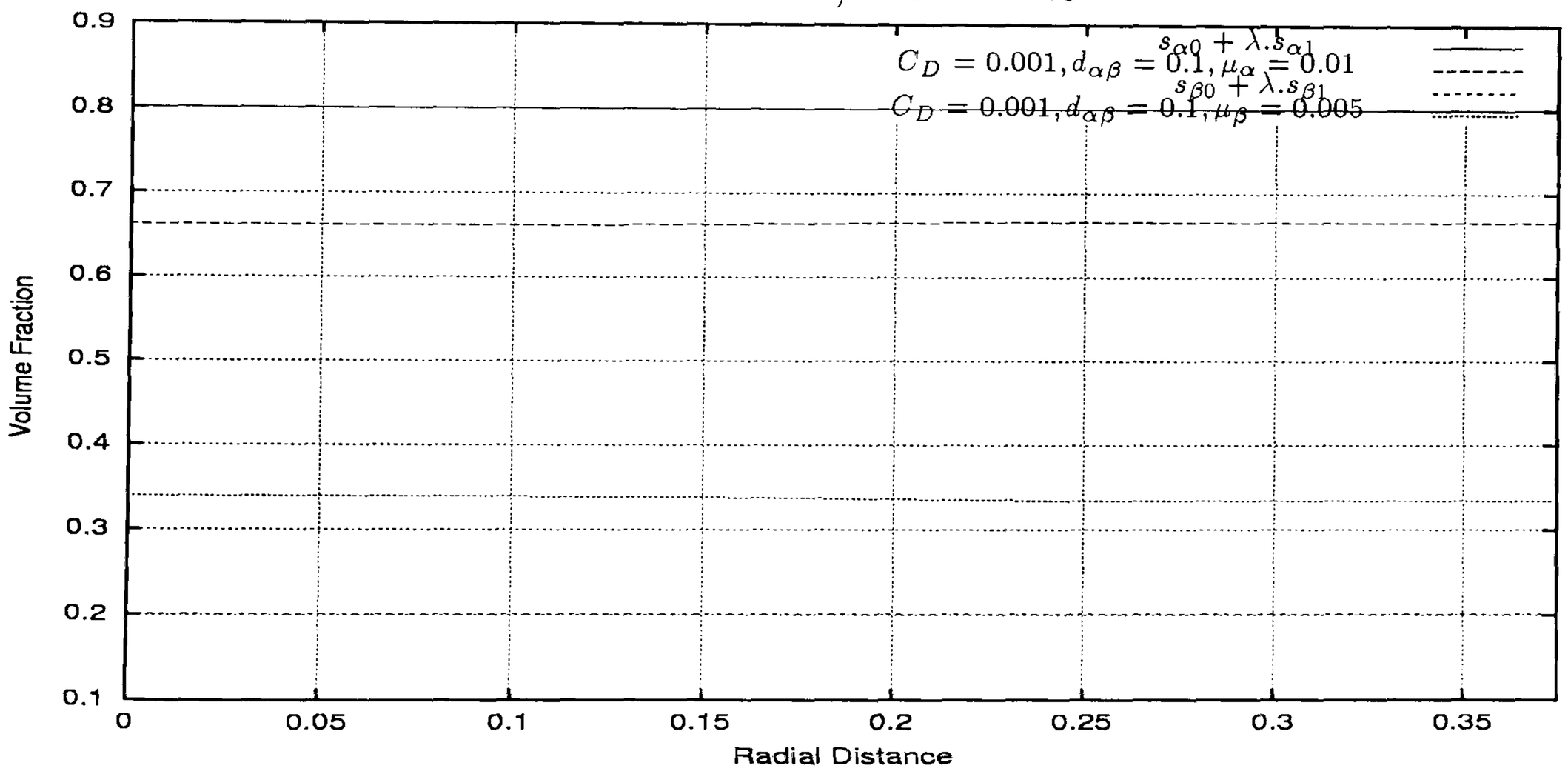


Figure 6.69: Two-phase flow through a Cylinder - comparison between the analytical volume fraction and the computational one along the line $z = 12.0$ for $\lambda = 0.01$. It can be observed that the volume fractions computed analytically and computationally are constant.

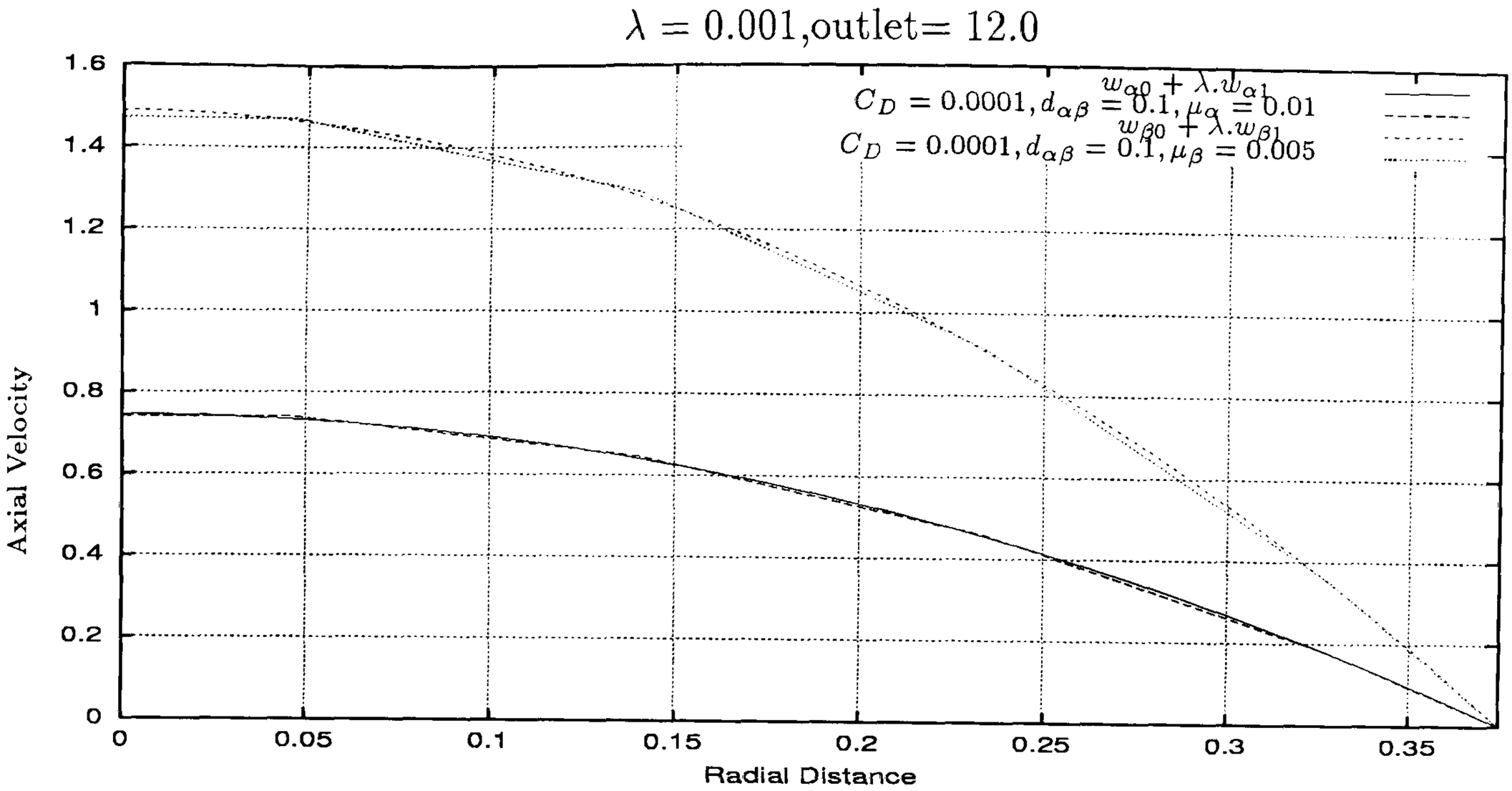


Figure 6.70: Two-phase flow through a Cylinder - comparison between the analytical axial velocity and the computational one along the line $z = 12.0$ for $\lambda = 0.001$. It can be observed excellent agreement between the analytical axial velocities and the computational ones for both phases

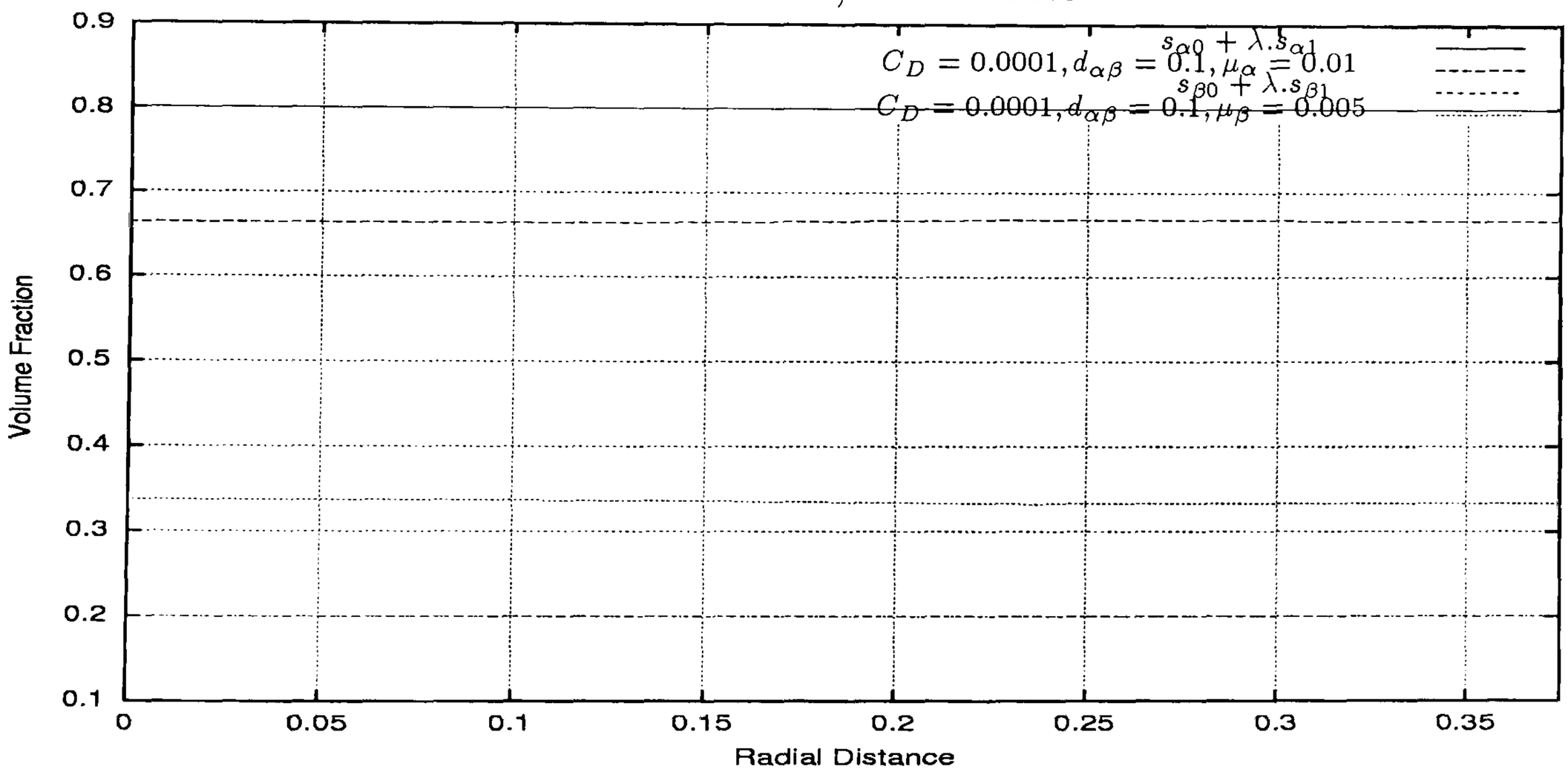


Figure 6.71: Two-phase flow through a Cylinder - comparison between the analytical volume fraction and the computational one along the line $z = 12.0$ for $\lambda = 0.001$. It can be seen that the volume fractions computed by both ways are constant.

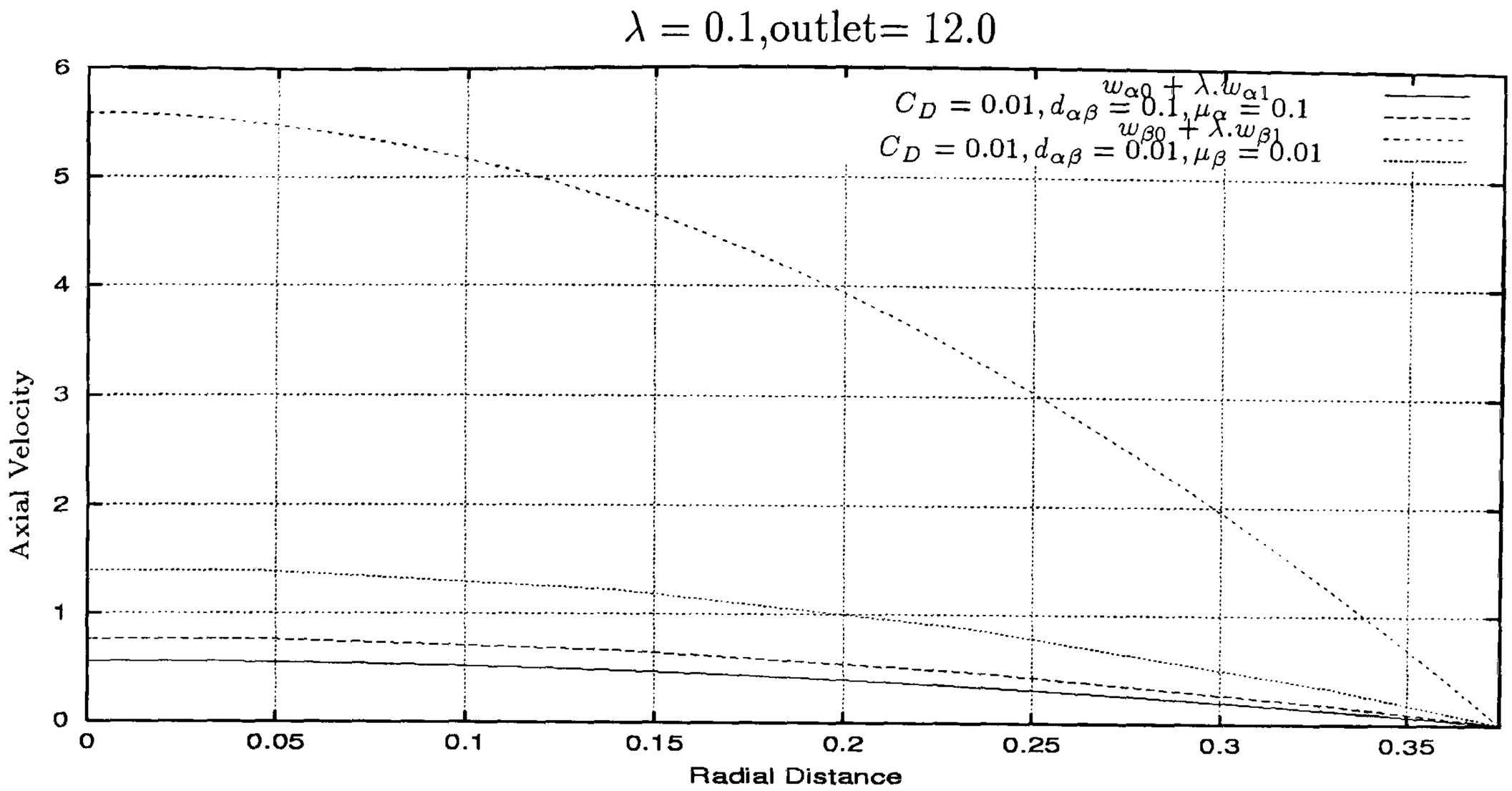


Figure 6.72: Two-phase flow through a Cylinder - comparison between the analytical axial velocity and the computational one along the line $z = 12.0$ for $\lambda = 0.1$. It can be seen that phase α profiles exhibit a general good agreement while phase β profiles some disagreement.

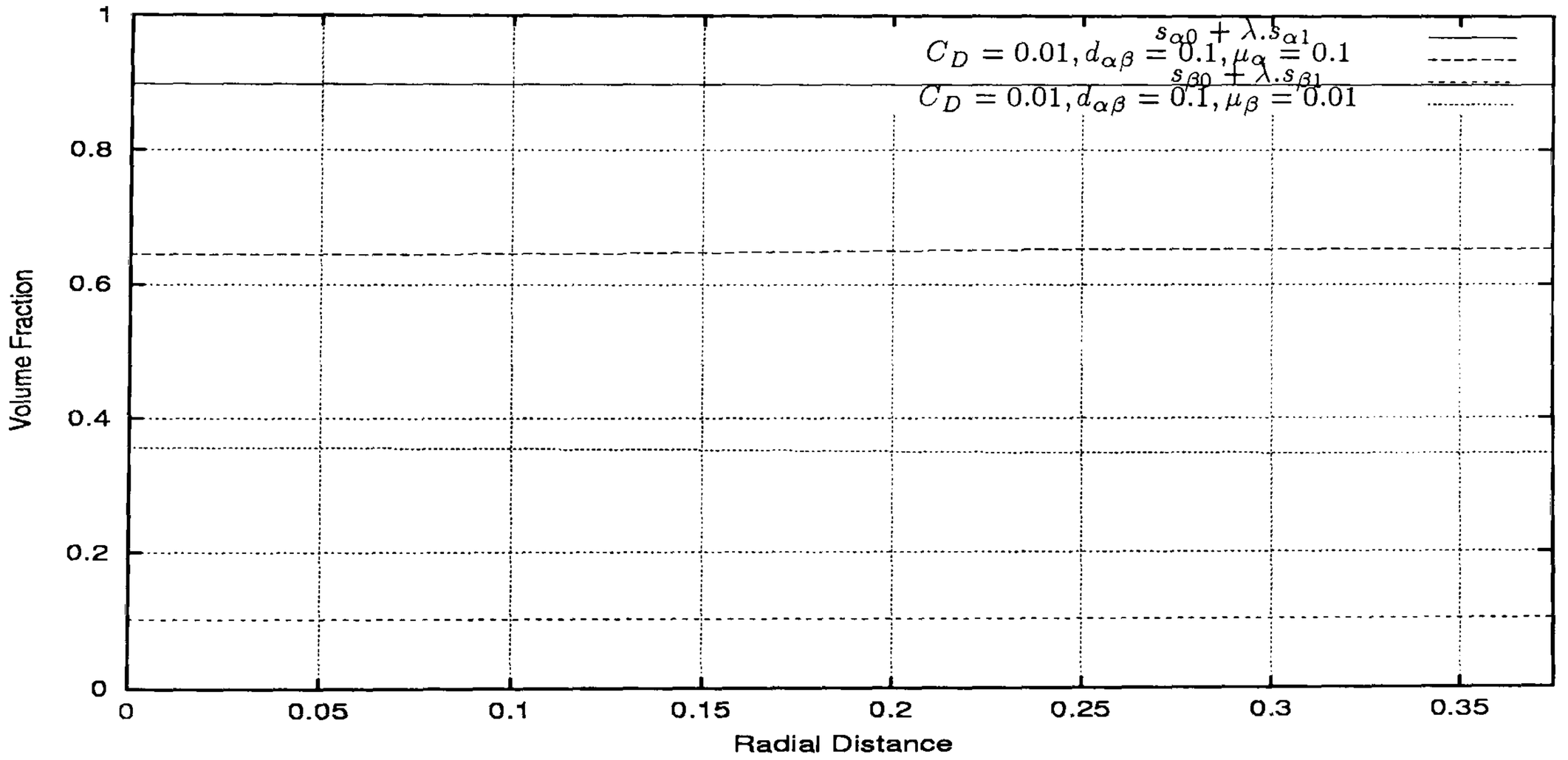


Figure 6.73: Two-phase flow through a Cylinder - comparison between the analytical volume fraction and the computational one along the line $z = 12.0$ for $\lambda = 0.1$. It can be observed that the volume fractions computed analytically and computationally are constant.

6.3.6 Coupling of the One-Dimensional Solver Emaps with CFX 4.3

In this section we are going to couple the one-dimensional solver **Emaps** with the commercial code **CFX 4.3** in a two-phase fluid flow through a pipe with a contraction. The **CFX 4.3** code is used with the axial symmetry option, so this test case is basically the coupling of a one-dimensional code with a two-dimensional one. One important difference between the test case discussed in Section 6.3.3 is that the two-dimensional solver works in cylindrical coordinates. In fact one limitation that we had with **transient Pamg Multi-phase** was precisely the fact of working only with Cartesian coordinates. The way in which the information between the different dimensional regions is interchanged is similar to the way used in the first version of the **D-adaptive solver**. Each dimensional region produces an output file that is used as input file in the computation of the consecutive dimensional region. The geometry of this test case is presented in Figure 6.74. In Figure 6.75 we see the sections where the solution profiles are going to be presented.

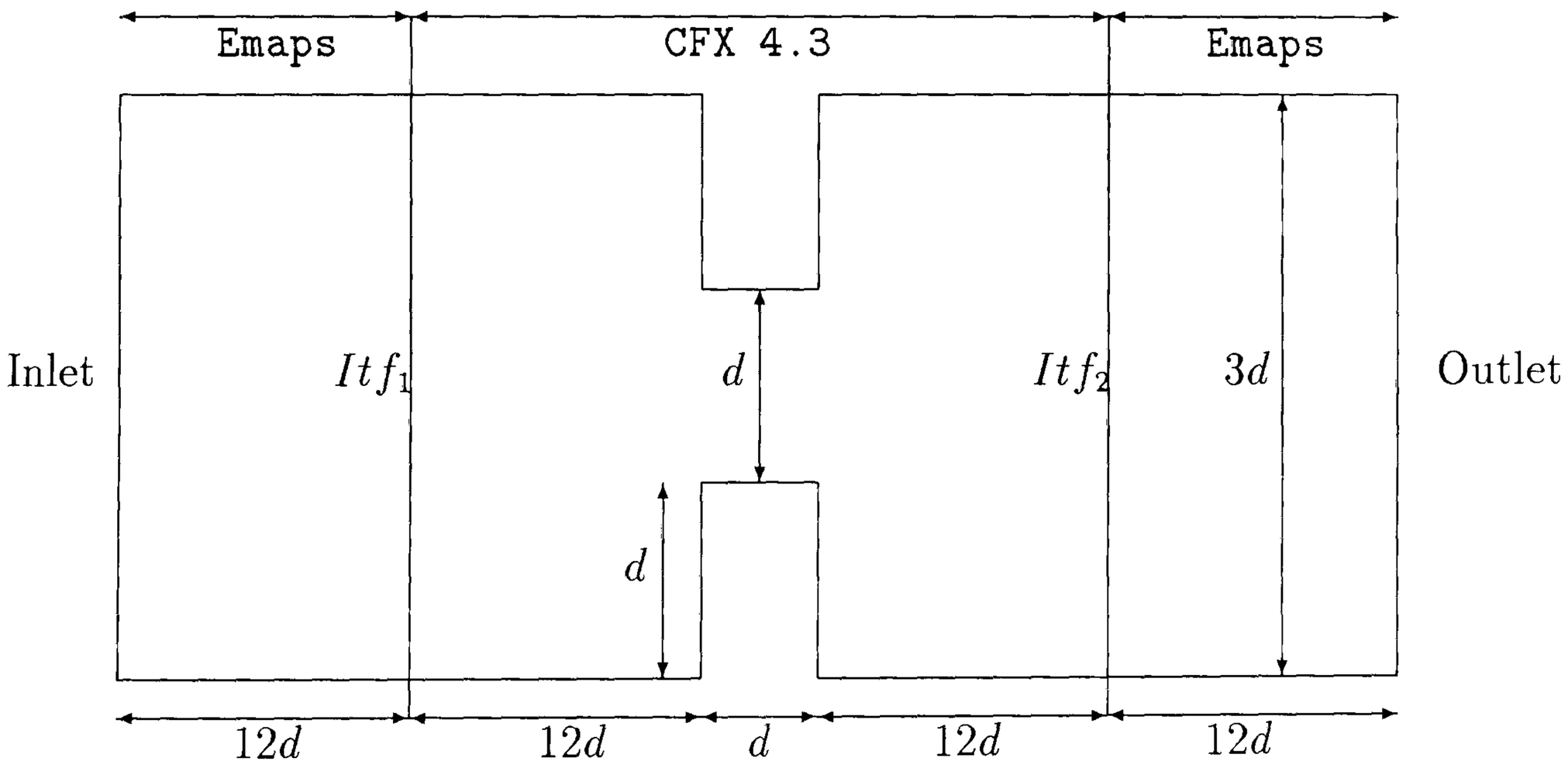


Figure 6.74: Geometrical representation of the domain of the pipe-contraction flow problem

The boundary conditions used are: In the inlet :

- Velocities $u(z) = 0.1$ for both phases
- Volume fractions $r(z) = 0.5$ for both phases

At the outlet : Neumann conditions.

The value chosen for the drag coefficient is $C_D = 0.0$. Hence the values chosen for the inter-facial length are irrelevant.

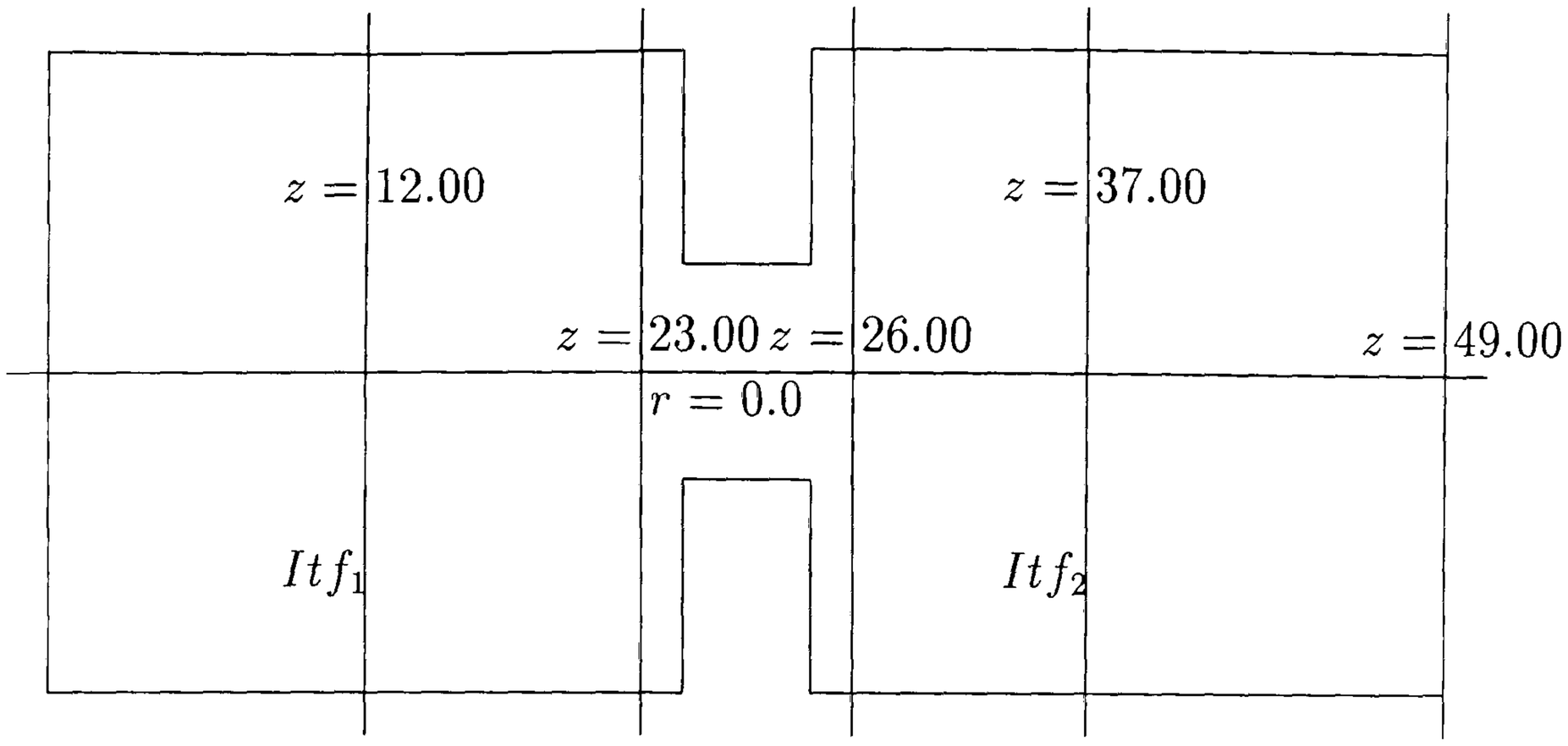


Figure 6.75: Pipe-contraction flow problem - main sections to give the solution profiles

The physical properties chosen for this test case are:

Two-Phase Flow	Reynolds Number	Viscosity	Density
Phase α	7.5	0.01	1.0
Phase β	15	0.005	1.0

The treatment of the interface Itf_1 (see Figure 6.74) is similar to the treatment that we have specified for the `D-adaptive solver` constructed with `Emaps` and `Pamg`.

Using Equations (6.85) and (6.86) derived in Section 6.3.4

$$\begin{aligned} w_{\alpha 0} &= \frac{1}{4\mu_\alpha}\frac{\partial p_0}{\partial z}(r^2 - R^2) \\ w_{\beta 0} &= \frac{1}{4\mu_\beta}\frac{\partial p_0}{\partial z}(r^2 - R^2) \ . \end{aligned}$$

we have to compute an approximation for $\frac{\partial p_0}{\partial z}$ which can be done through the expression

$$\frac{\partial p_0}{\partial z} = -\frac{8\mu_\alpha Q_\alpha}{\pi s_{\alpha 0} R^4} = -\frac{8\mu_\beta Q_\beta}{\pi s_{\beta 0} R^4} \ .$$

Representing by $\overline{u(z)}$ and $\overline{r(z)}$ the velocity and volume fractions for each phase computed by the one-dimensional code at point $z = 12.0$, the fluxes Q_α and Q_β are approximated by the expression $\overline{u(z)} \times \overline{r(z)}$ which gives the flux computed for each phase by the one-dimensional solver.

In Interface Itf2 the interpolation formulas used for a uniform grid are:

$$\overline{u(z)} = \frac{1}{R}\int_0^R wrdr = \frac{1}{R}\lim_{n\to\infty}\sum_{i=1}^{n} w_i r_i \Delta r_i \approx \frac{\sum_{i=1}^{n} w_i}{n}$$

and

$$\overline{r(z)} = \frac{1}{R}\int_0^R srdr = \frac{1}{R}\lim_{n\to\infty}\sum_{i=1}^{n} s_i r_i \Delta r_i \approx \frac{\sum_{i=1}^{n} s_i}{n}$$

where w_i, s_i are the axial velocity and volume fractions at the cell i and n is the number of cells.

Figures 6.76 and 6.77 present the comparison between the axial velocity computed by `D-adaptive solver` and `CFX 4.3` along the line $r = 0.1875$. The profiles for *Phase* α show a good overall agreement. However some slight differences can be observed. One is in the contraction region. The `D-adaptive solver` gives a bigger axial velocity in the middle of the contraction. Moreover when changing back to the one-dimensional region the velocity falls a little. This must be due to the interpolation errors during treatment of the interface Itf_2. This same behaviour can be observed in the figure corresponding to *Phase* β (Figure 6.77). A curious feature that can be observed in this figure is the small slope down that can be observed between 0.0 and 5.0 in the profile of `CFX 4.3` and between 13.0 and 17.0 by the `D-adaptive solver`. Hence both solvers verify this `CFX 4.3` peculiarity. Equivalent features can be observed in Figures 6.78 and 6.79 which show the comparison between the volume fractions for phases α and β along the line $r = 0.1875$ for both solvers. In Figure 6.80 we can observe the pressure drop profile along the line $r = 0.1875$ produced by `D-adaptive solver` and `CFX 4.3`. The only disagreement observed is between the profile corresponding to the one-dimensional regions and a slightly different inclination of the lines corresponding to the geometry constraint.

In Figure 6.81 we present the comparison between the axial velocities for phases α and β along the line $z = 12.0$ produced by `D-adaptive solver` and `CFX 4.3`. We can see the good agreement obtained in the results plotted by the two solvers. The velocities of *Phase* α are approximately the double of the velocities of *Phase* β. This is exactly what we should expect due to the relation between the viscosities of the fluids involved (see Equations (6.85) and (6.86)).

Figures 6.82 and 6.83 present the results for the axial and radial velocities along the line $z = 23.0$. The axial velocities produced by both solvers are almost identical. Figure 6.83 shows that the radial velocities are null over all the radial distance.

Figures 6.84, 6.85 and 6.86 present the results for the axial velocity, radial velocity and volume fraction for phases α and β along the line $z = 26.0$. In this line which is situated after the contraction is visible that the radial velocity for *Phase* α is small but not null and the relation between the axial velocities is much bigger. But general agreement between the profiles produced by both solvers is verified.

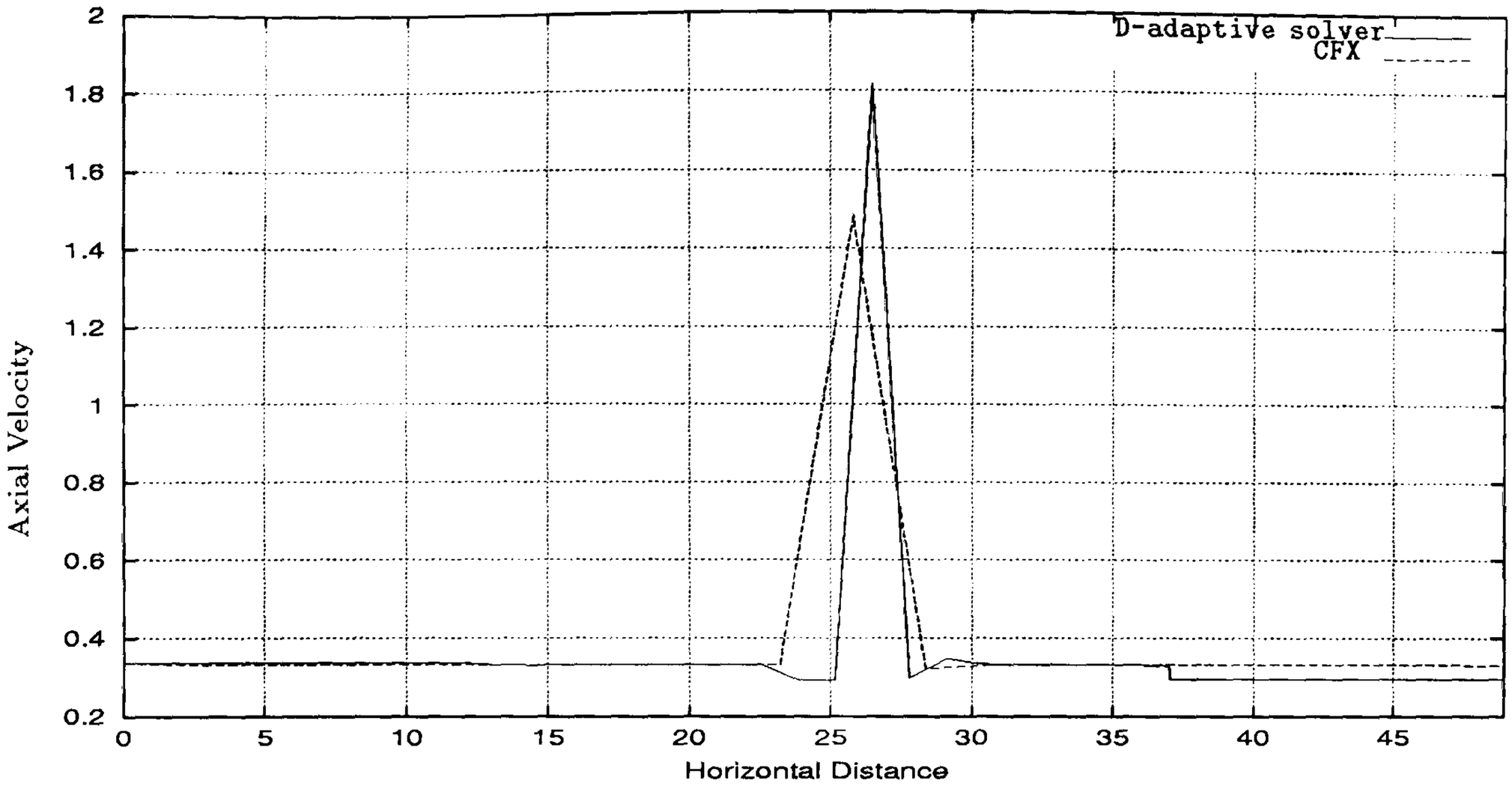


Figure 6.76: Two-phase flow through a Contraction in a Pipe - comparison between the axial velocity profile for *Phase* α along the line $r = 0.1875$ for `D-adaptive solver` and `CFX 4.3`. It can be seen that `D-adaptive solver` gives a bigger axial velocity in the middle of the contraction and produces a little jump due to interpolation errors on the two-dimensional/one-dimensional interface.

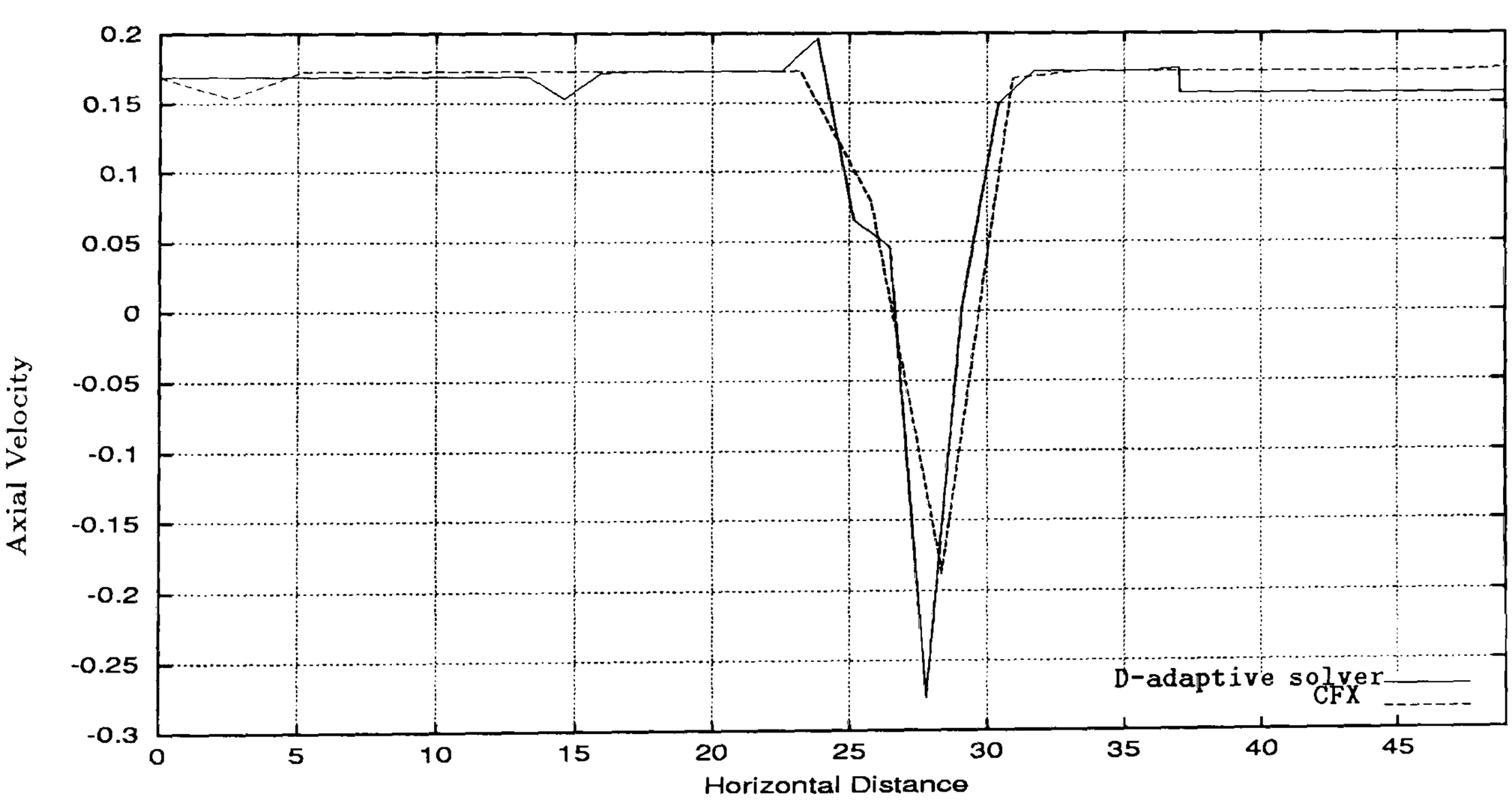


Figure 6.77: Two-phase flow through a Contraction in a Pipe - comparison between the axial velocity profile for *Phase* β along the line $r = 0.1875$ for `D-adaptive solver` and `CFX 4.3`. It can be seen that `D-adaptive solver` gives a smaller axial velocity in the middle of the contraction and produces a little jump due to interpolation errors on the two-dimensional/one-dimensional interface. Note the little slope that `CFX 4.3` produce in the beginning of the domain, which is propagated to the `D-adaptive solver` when it starts working in the two-dimensional region.

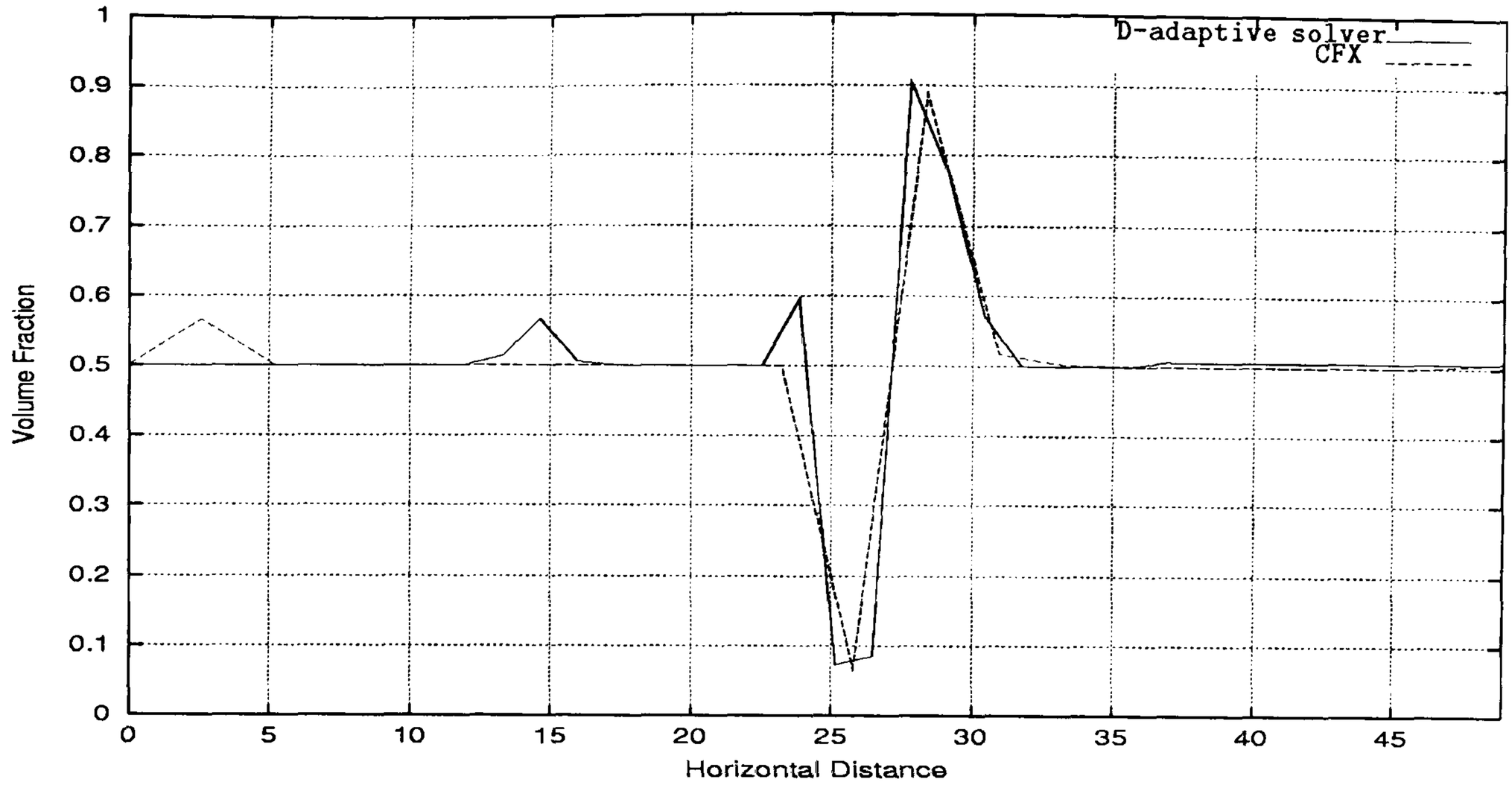


Figure 6.78: Two-phase flow through a Contraction in a Pipe - comparison between the volume fraction profile for *Phase* α along the line $r = 0.1875$ for `D-adaptive solver` and `CFX 4.3`. It can be seen good general agreement between the results produced by both solvers. Note the little slope that `CFX 4.3` produce in the beginning of the domain which is propagated to the `D-adaptive solver` when it starts working in the two-dimensional region.

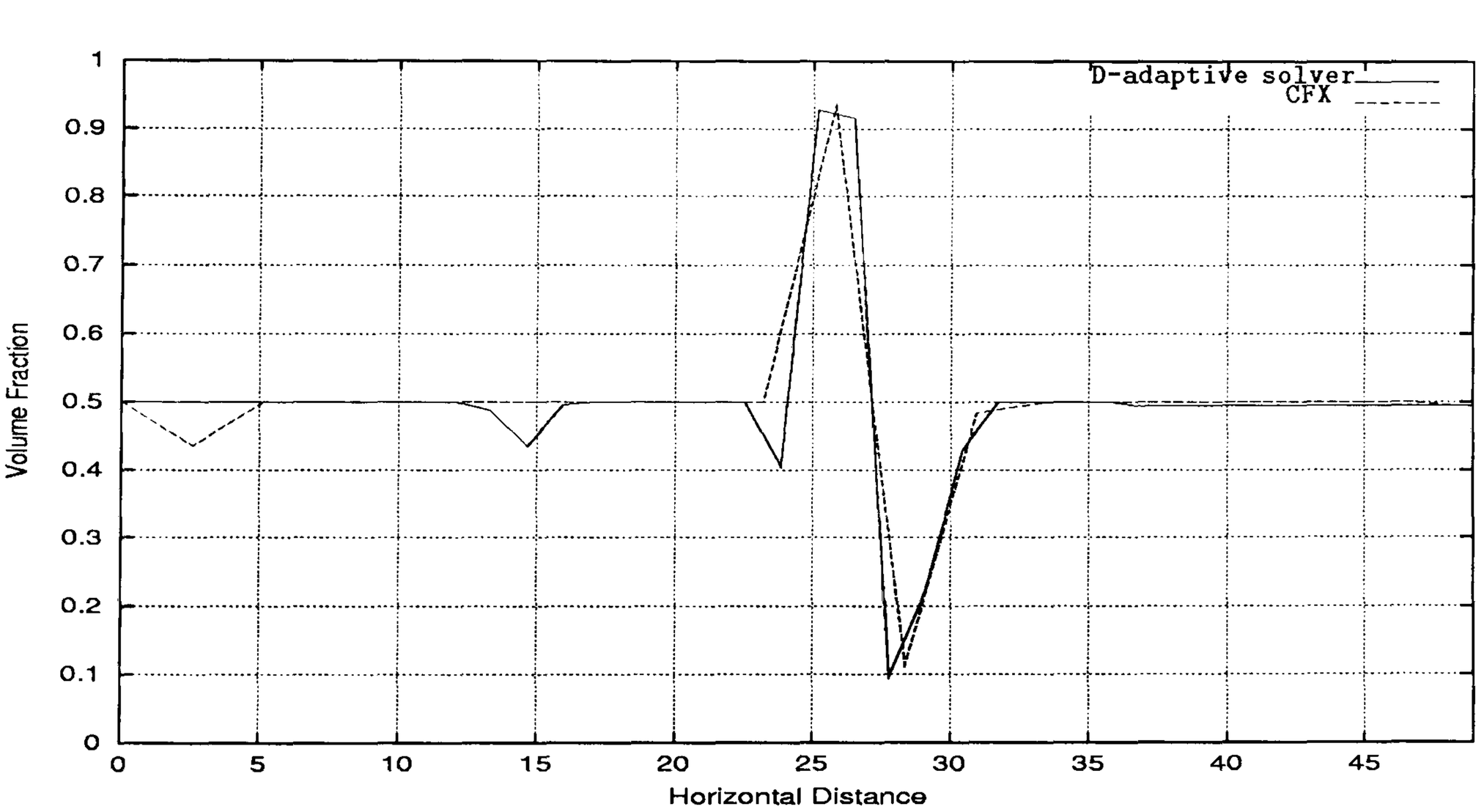


Figure 6.79: Two-phase flow through a Contraction in a Pipe - comparison between the volume fraction profile for *Phase* β along the line $r = 0.1875$ for `D-adaptive solver` and `CFX 4.3`. It can be seen good general agreement between the results produced by both solvers. Note the little slope that `CFX 4.3` produce in the beginning of the domain, which is propagated to the `D-adaptive solver` when it starts working in the two-dimensional region.

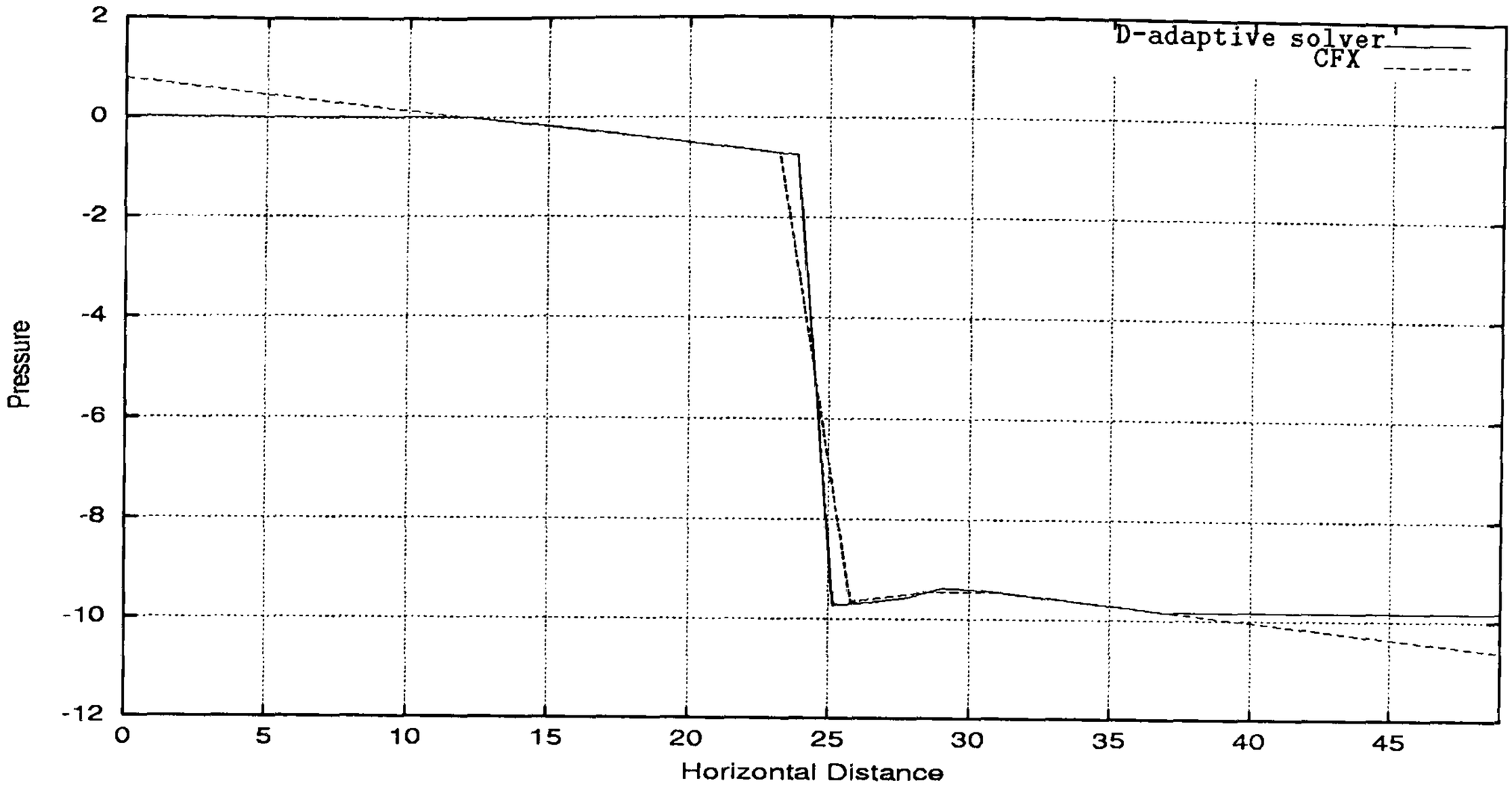


Figure 6.80: Two-phase flow through a Contraction in a Pipe - comparison between the pressure drop profile along the line $r = 0.1875$ for `D-adaptive solver` and `CFX 4.3`. It can be seen slightly different inclination of the lines corresponding to the geometry constraint and no parallel straight lines in the one-dimensional regions.

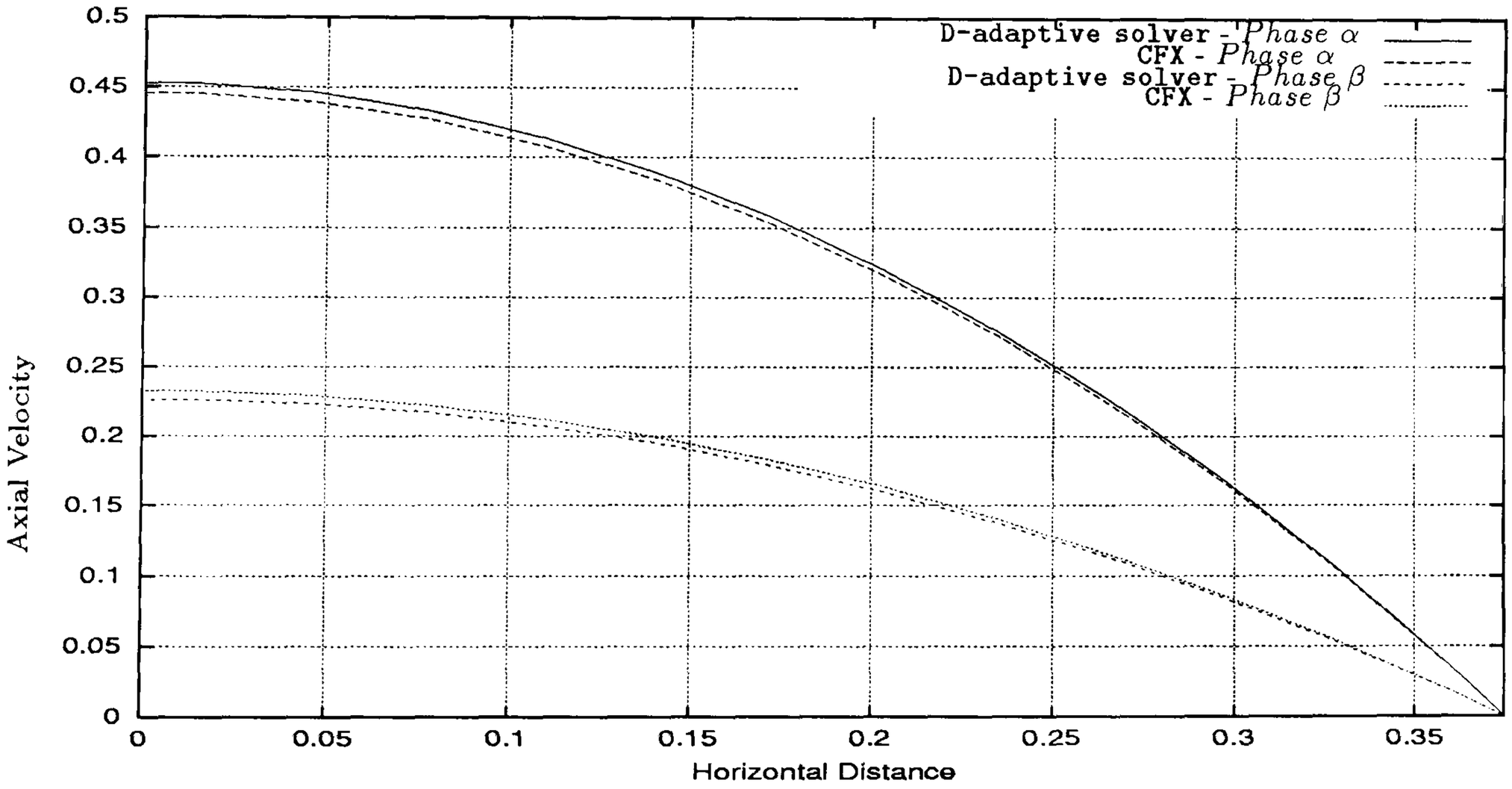


Figure 6.81: Two-phase flow through a Contraction in a Pipe - comparison between the axial velocity profile along the line $z = 12.0$ for `D-adaptive solver` and `CFX 4.3`. It can be seen good agreement between the velocities produced by both solvers.

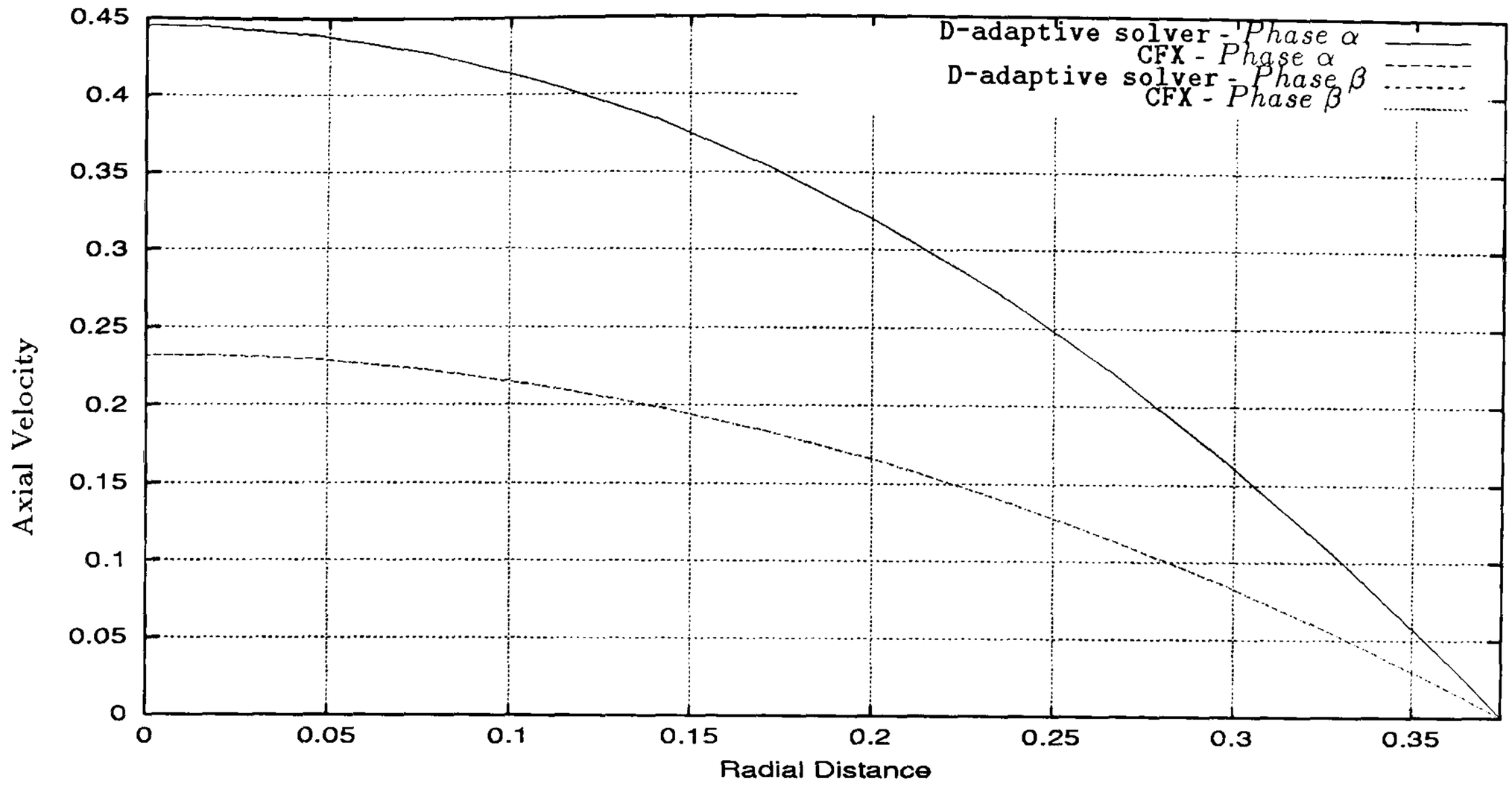


Figure 6.82: Two-phase flow through a Contraction in a Pipe - comparison between the axial velocity profile along the line $z = 23.0$ for `D-adaptive solver` and `CFX 4.3`. It can be observed good agreement between the axial velocities produced by both solvers.

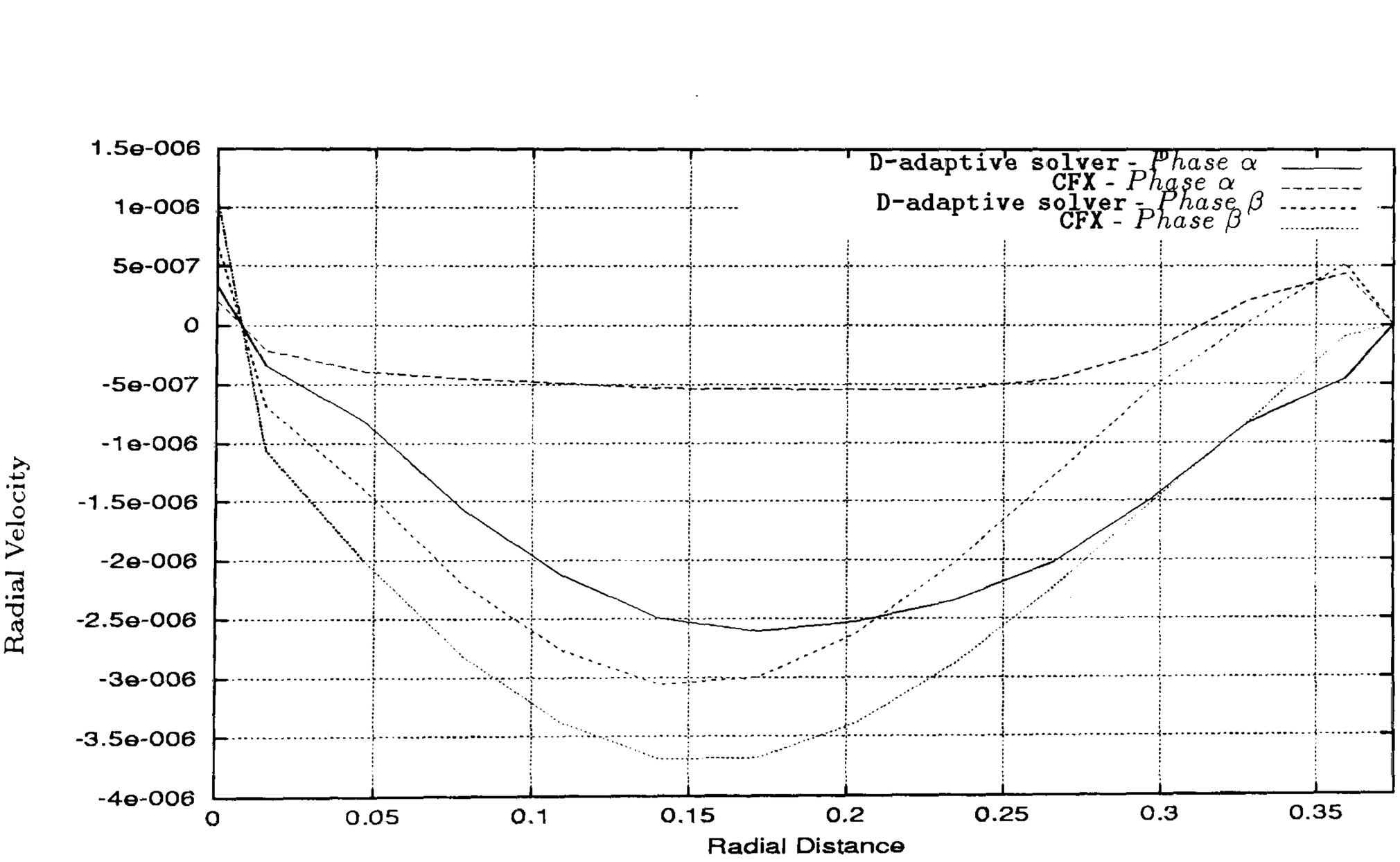


Figure 6.83: Two-phase flow through a Contraction in a Pipe - comparison between the radial velocity profile along the line $z = 23.0$ for `D-adaptive solver` and `CFX 4.3`. It can be seen that the radial velocities produced by both solvers are close to null. Note the scale of the radial velocity axis.

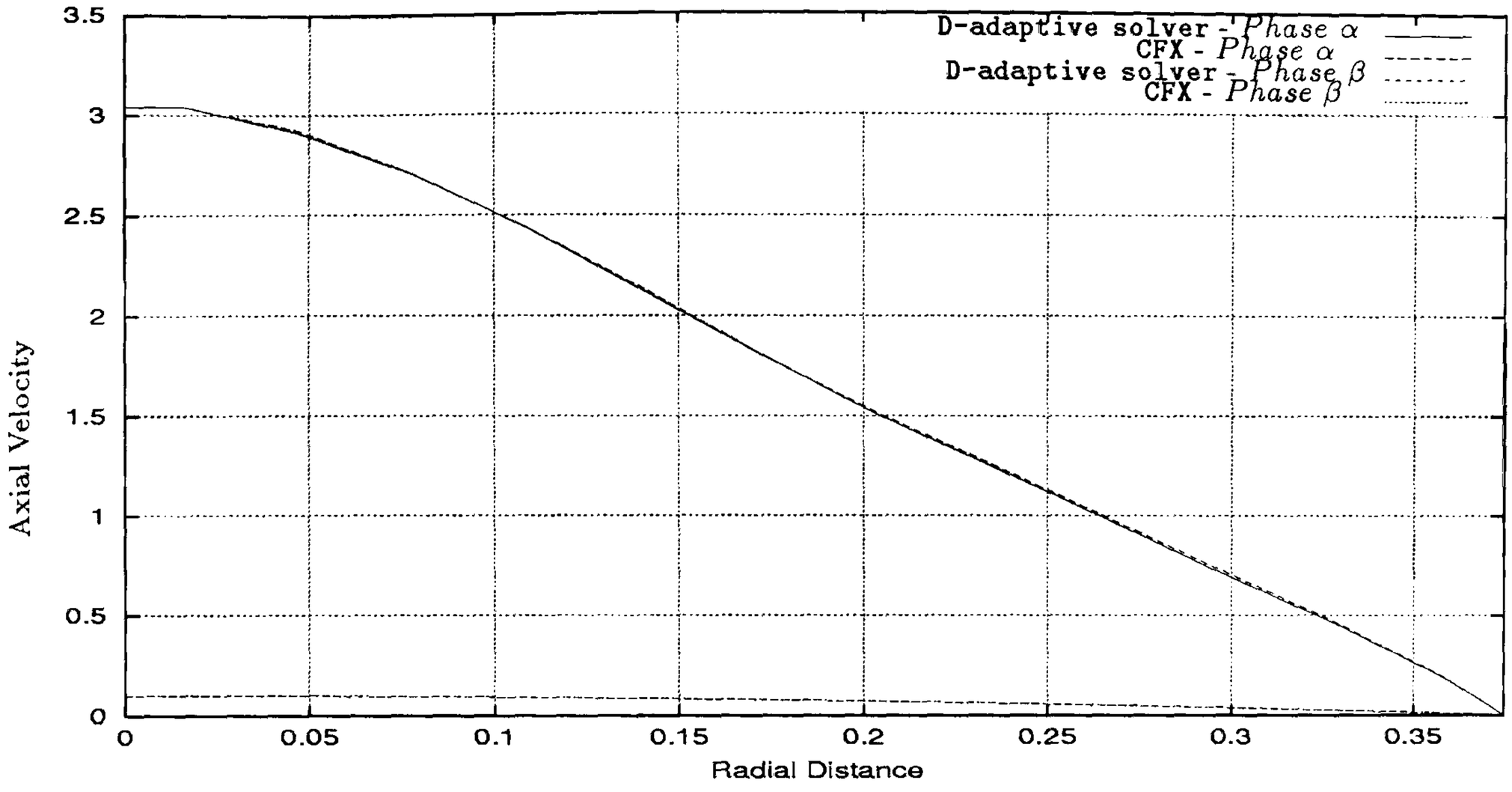


Figure 6.84: Two-phase flow through a Contraction in a Pipe - comparison between the axial velocity profile along the line $z = 26.0$ for `D-adaptive solver` and `CFX 4.3`. It can be seen good agreement between the axial velocities computed by both solvers.

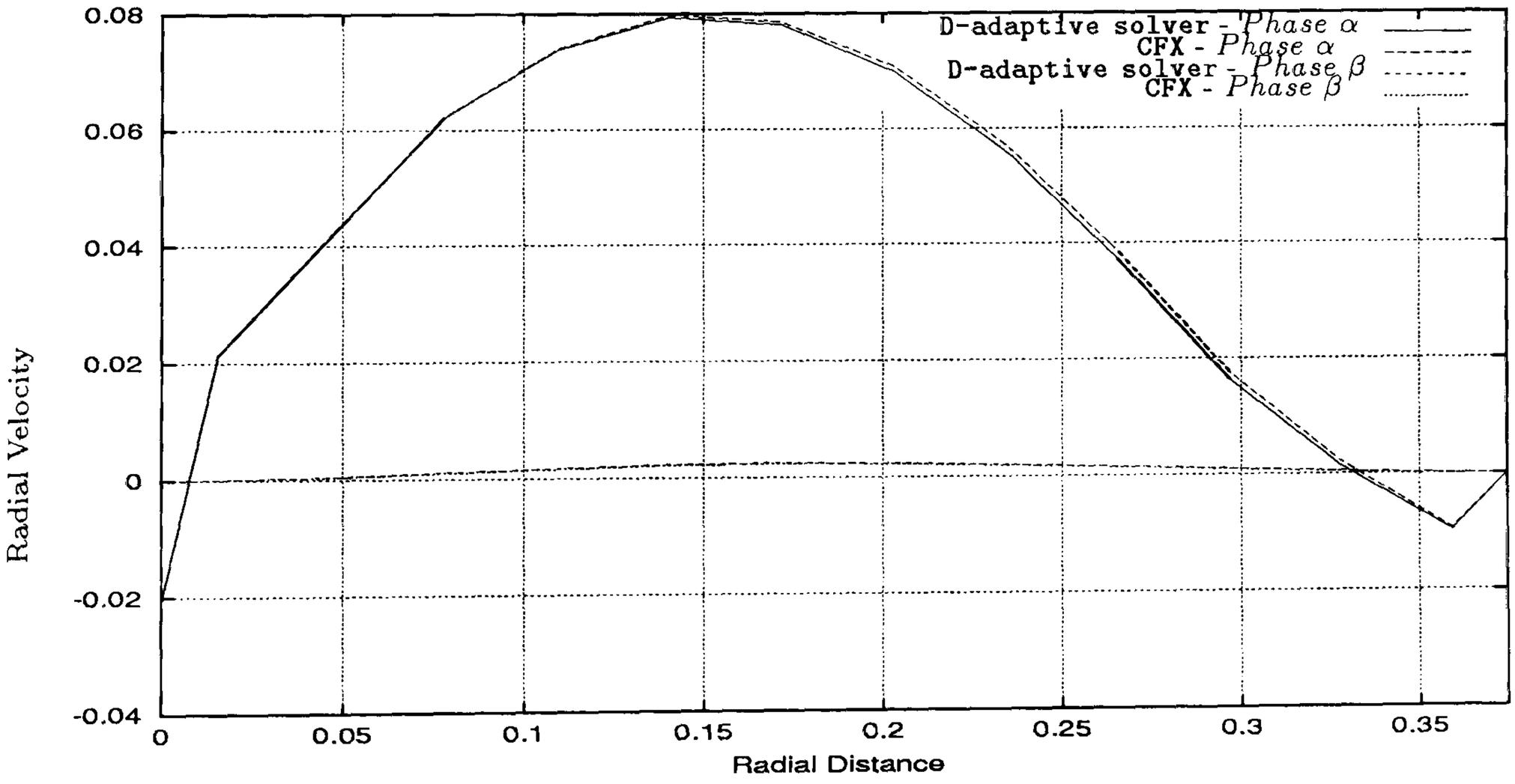


Figure 6.85: Two-phase flow through a Contraction in a Pipe - comparison between the radial velocity profile along the line $z = 26.0$ for `D-adaptive solver` and `CFX 4.3`. It can be seen good agreement between the results produced by both solvers. Note the negative values of phase α near the walls.

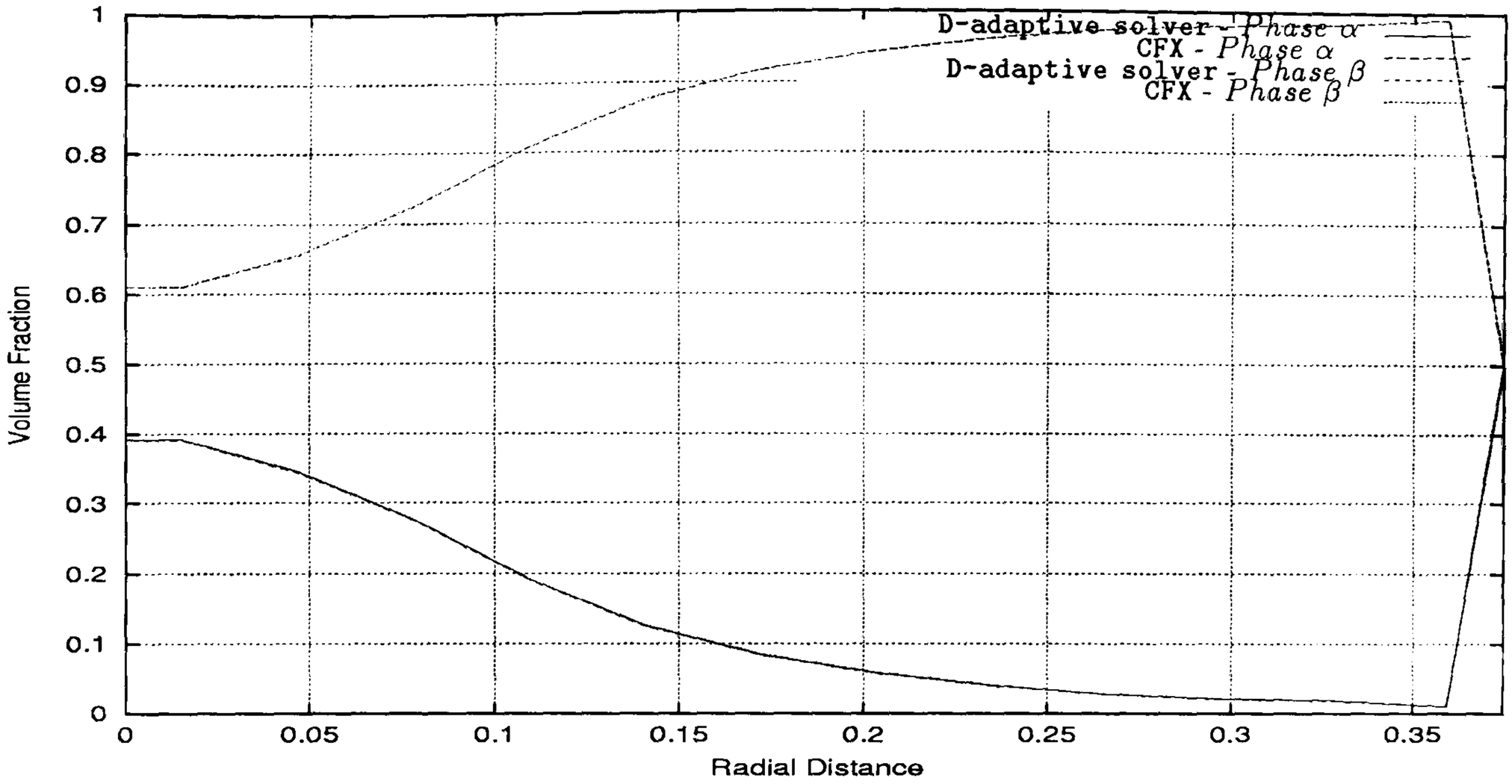


Figure 6.86: Two-phase flow through a Contraction in a Pipe - comparison between the volume fractions profile along the line $z = 26.0$ for `D-adaptive solver` and `CFX 4.3`. It can be observed good general agreement between the profiles produced by both solvers.

Figures 6.87, 6.88 and 6.89 present the results for the axial velocity, radial velocity and volume fraction for phases α and β along the line $z = 37.0$. In this second interface line we can see the parabolic profile exhibit by the axial velocities of both phases in both solvers with once more an excellent agreement with both results. Figure 6.88 clearly evidences the null velocity of the radial velocity produced by both solvers. Finally in Figure 6.89 we can observe the volume fraction profiles. Although we can see some disagreement in the profiles, note the small scale of the volume fraction axis.

In the outlet the values are present in Table 6.3. This has been done by integrating the two-dimensional results produced by `CFX 4.3` over the radius. It is clear that the values produced by both solvers are very close.

Finally in Table 6.4 we present the CPU-time spent by `D-adaptive solver` and `CFX 4.3` in the simulation of a two-phase flow through a pipe with a contraction.

The `D-adaptive solver` for this problem can save about 65% of the CPU-time spent by the `CFX 4.3` solver.

In this section we validated our second version of the `D-adaptive solver`. This version has been constructed coupling the one-dimensional solver `Emaps` and `CFX 4.3` solver used as a two-dimensional solver. The validation has been done by comparing the results produced by this `D-adaptive solver` with the results produced by `CFX 4.3` in a test problem. This has been the simulation of a two-phase fluid flow trough a pipe with a contraction. Observing the results produced by `CFX 4.3` over all the entire domain of the problem decision has been taken

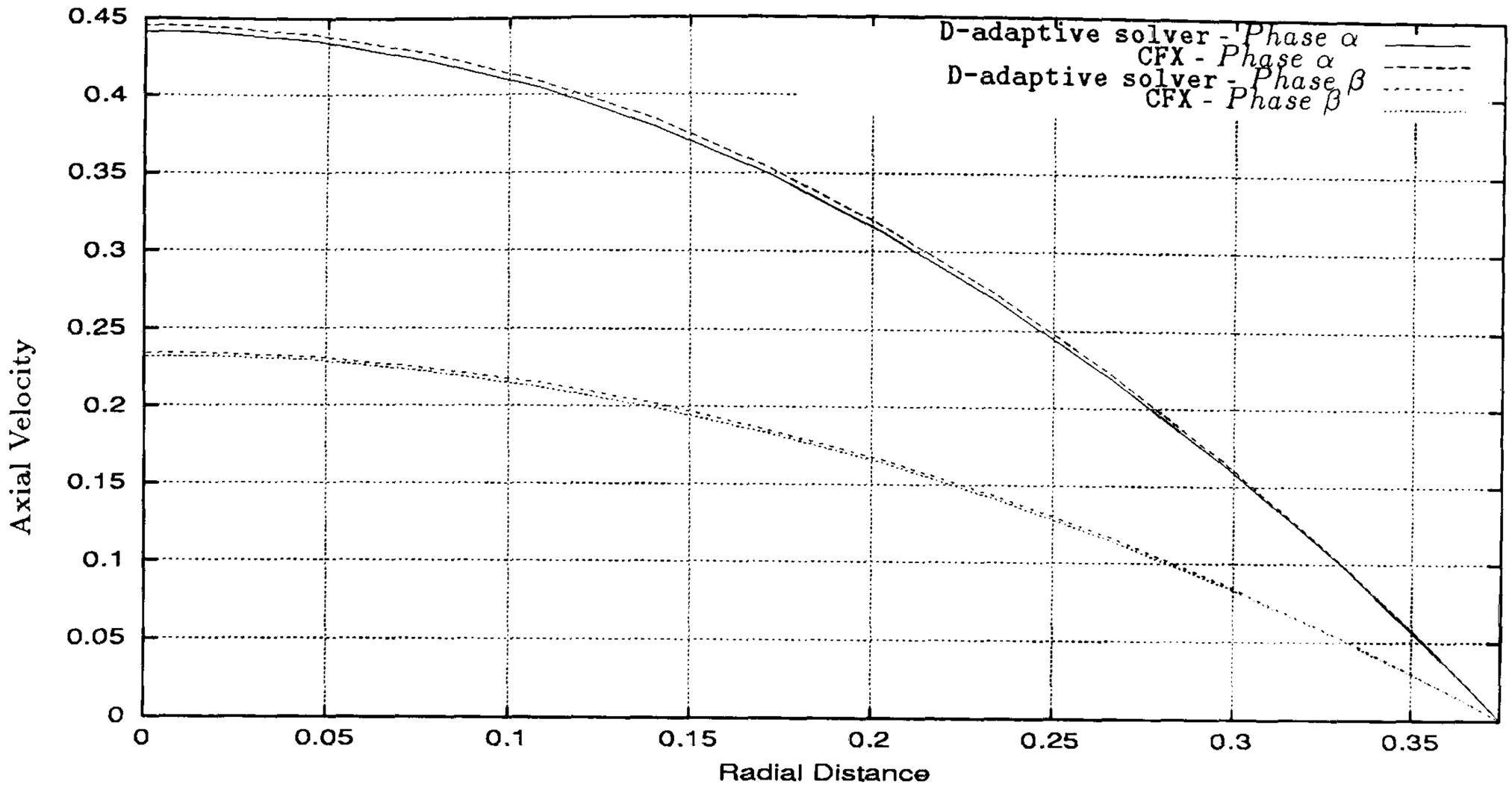


Figure 6.87: Two-phase flow through a Contraction in a Pipe - comparison between the axial velocity profile along the line $z = 37.0$ for `D-adaptive solver` and `CFX 4.3`. It can be seen good agreement between the axial velocities computed by both solvers.

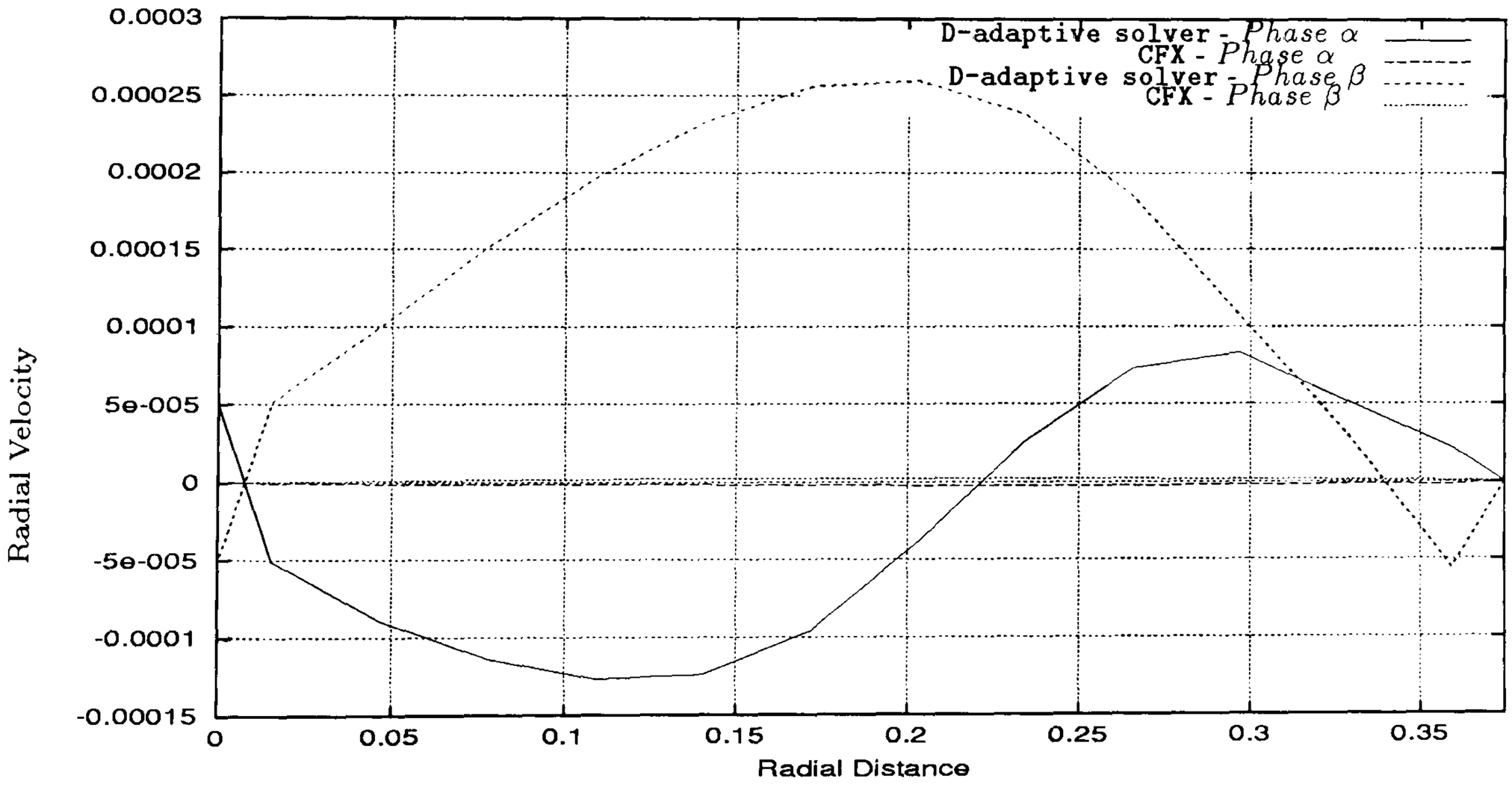


Figure 6.88: Two-phase flow through a Contraction in a Pipe - comparison between the radial velocity profile along the line $z = 37.0$ for `D-adaptive solver` and `CFX 4.3`. It can be seen that the radial velocities are very close to null. Note the small scale of the radial velocity axis.

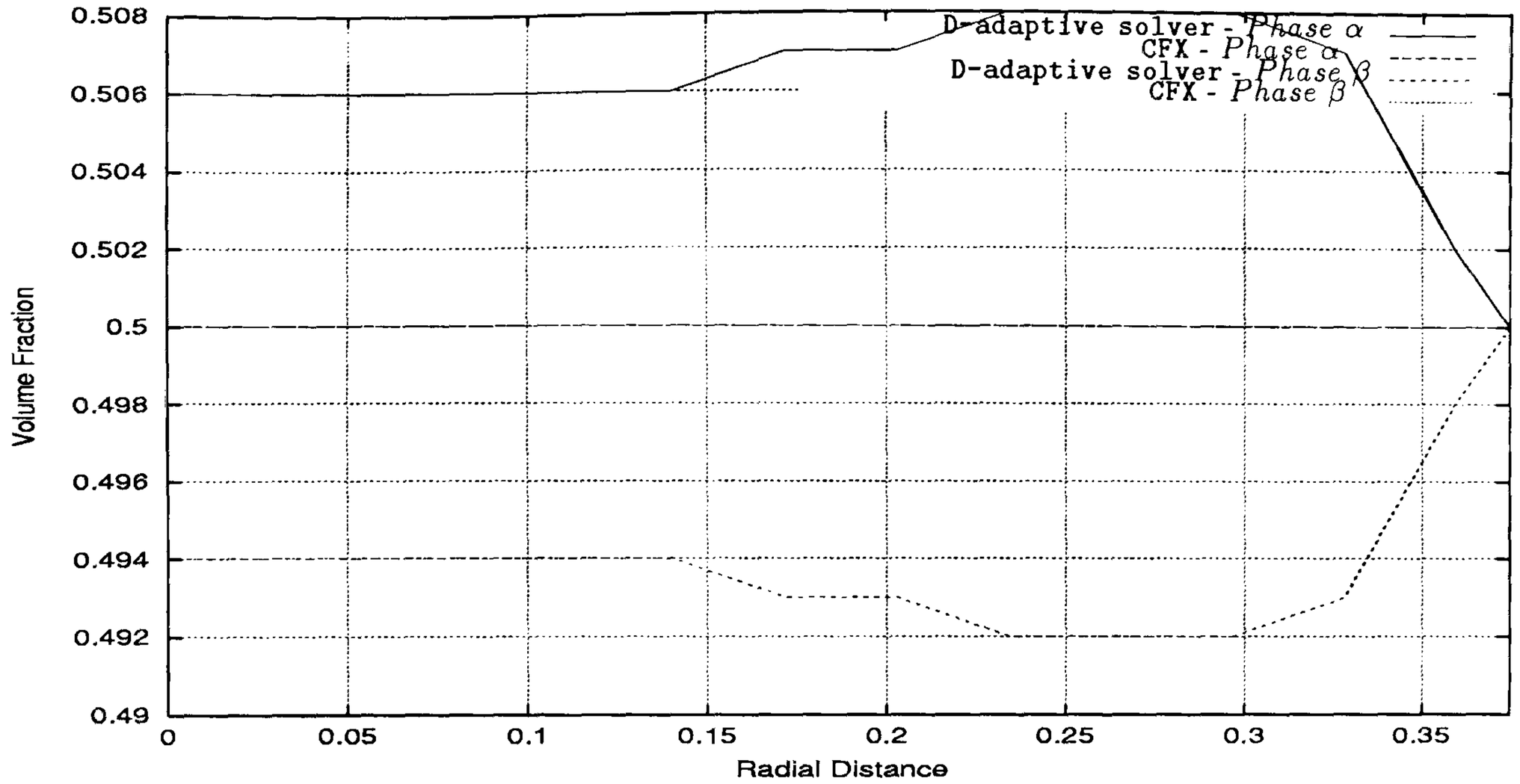


Figure 6.89: Two-phase flow through a Contraction in a Pipe - comparison between the volume fractions profile along the line $z = 37.0$ for `D-adaptive solver` and `CFX 4.3`. It can be observed good general agreement between the volume fractions produced by both solvers. Note the small scale of the volume fraction axis.

about the placement of the dimensional interfaces. In these interfaces formulas derived and validated in the preceding sections have been used. Naturally the set of fluids used in the simulation has been chosen to satisfy the asymptotic formulas derived and validated in the two previous sections. It has been verified that with this `D-adaptive solver` we can save about 65% of the CPU-time spent by `CFX 4.3` when solving the same problem in the entire domain. Additional gains in the performance can be think of if the geometry of the problem is bigger than the geometry of our test problem. The general agreement of the results has been verified. However some small differences have been observed. The most relevant one respects to the pressure drop issue. Good agreement has been observed in the contraction region. However in the one-dimensional regions the pressure drop produced by `D-adaptive solver` and `CFX 4.3` is not exactly the same. However, the differences are lower than those one observed in the validation of our first version of the `D-adaptive solver`. Another slight difference that could be observed was the little jumps on the `D-adaptive solver` curves due to the interpolation errors on the two-dimensional/one-dimensional interfaces. However we could conclude that there was a general good agreement between the solutions produced by both solvers. For instance it has been verified that at the outlet of the domain the velocities and volume fractions computed by both solvers where very similar.

Two-Phase Flow	D-adaptive solver	CFX 4.3
Velocity - Phase α	0.295000	0.294850
Velocity - Phase β	0.157000	0.156342
Volume Fraction - Phase α	0.506000	0.506250
Volume Fraction - Phase β	0.494000	0.493750

Table 6.3: Two-phase flow through a Contraction in a Pipe - comparison between the velocities and volume fractions along the line $z = 49$ for D-adaptive solver and CFX 4.3. It can be seen that the values produced by both solvers are very close.

Two-Phase Flow	D-adaptive solver	CFX 4.3
First One-dimensional domain	8 sec	
Two-dimensional domain	9447 sec	
Second One-dimensional domain	24 sec	
Overall domain	9479 sec	27390

Table 6.4: CPU-time spent by the D-adaptive solver and the CFX 4.3 in the simulation of the contraction two-phase flow in a pipeline

6.4 Conclusions

In this Chapter we have presented some novel techniques for the solution of multi-phase fluid flows through complex geometries. These new techniques are based on the concept of D-adaptivity. This dimensional adaptivity has been implemented in two difficult problems of a two-phase flow through a contraction. These problems have been chosen because of its very complex geometry and their frequent occurrence in the Computational Fluid Dynamics field. The characteristics of geometry, namely the length, with and location of the constraints, have been implemented in order to validate the essential results of each solver. One limitation that we had in this project was quite naturally the computational time. Hence the choice of restricted domains had to take place. However the `D-adaptive solver` can be easily implemented in other geometries with different characteristics.

The first step of the discussion on the dimensional coupling started with the derivation of the formulas to apply in the interface of a one-dimensional single-phase solver with a two-dimensional single-phase one. The Navier-Stokes equations for a one-dimensional fluid flow in a channel have been the starting point for this approximation. New interpolation formulas for the interface region have been derived. One important conclusion has been assessed: the pressure drop measured through the channel is constant and consequently the value of the pressure is only a function of the direction of the unidirectional flow.

After having established the interpolation formulas for the single-phase fluid flow the focus has been on the implementation of similar interpolation formulas for the two-phase fluid flow case. Analytic expressions for the velocities and volume fractions have been achieved from an asymptotic expansion based on a small resistance coefficient. Some assumptions have been undertaken in order to simplify the governing multi-fluid differential equations.

The quality of this analytic formulation was verified by comparing its results with the results produced by the `transient Pamg Multi-phase` solver for a two-phase fluid flow through a channel. It has been verified that the quality of the approximation becomes better when the resistance coefficient becomes smaller. Furthermore the length of the channel is very important for the quality of the volume fractions results. Several experiments were taken with different pipe lengths that proved this.

The first version of an `D-adaptive solver` has been produced between the coupling of the one-dimensional solver `Emaps` with the two-dimensional one `transient Pamg Multi-phase`. To validate this solver two experiments of a two-phase fluid through a contraction have been outlined. A comparison of the results produced by the `D-adaptive solver` with the results produced by the `transient Pamg Multi-phase` over all the domain of each experiment have been produced. Some good agreement between the plotted profiles has been observed. However some disagreements have also been seen. One important reason for this is on the different coordinate systems that `Emaps` and `transient Pamg Multi-phase` use. The impact of this reality can not be adequately measured. The interpolation errors,

which necessarily occur when changing from one-dimensional region to another plus the different coordinate systems, can originate substantial differences on the results. However one important conclusion that could be undertaken was that the `D-adaptive solver` constructed this manner could save about 90% on CPU-time.

In order to construct a new version of the `D-adaptive solver` that could work always with the same system of coordinates, new asymptotic formulas for the velocities and volume fractions in cylindrical coordinates have been derived. The idea was to obtain the adequate expressions to apply in the interface of a one-dimensional problem with an axial-symmetry problem. The process has been similar to the derivation of the asymptotic formulas in cartesian coordinates. New governing equations for two-phase fluid uni-directional flow in cylindrical coordinates have been derived. These originated the expressions for the velocities and volume fractions.

The new asymptotic formulas in cylindrical coordinates have been validated with the simulation of a two-phase fluid flow through a pipe. The process of validation has been based on the comparison of the computational results produced by `CFX 4.3` and the results calculated by the analytical expressions. Three important conclusions can be outlined on this comparison process. One is that when choosing the resistance parameter smaller the computed results and the analytically computed ones become more close. Another important conclusion is that the volume fractions exhibit worst agreement than the velocities. Finally the characteristics of the fluids have an important role too. Evidence suggested that if the two fluids are very different in terms of the density and viscosity the asymptotic formulation could become inaccurate. The reason for that relies on the assumption used in the derivation process that there was no vertical mixing and consequently the volume fractions of each phase should remain constant.

Finally a new version of the `D-adaptive solver` has been constructed and validated against the commercial code CFX 4.3 in a very complex problem. The domain of this problem is a long pipe with a contraction in the middle. The agreement between the results presented by this `D-adaptive solver` and the `CFX 4.3` solver is very good. Consequently we can conclude that this implementation is correct. The main conclusion that it can be taken with this experiment is that the `D-adaptive solver` is about 35% faster than the commercial code `CFX 4.3`. Hence this methodology of coupling one-dimensional solver with a two-dimensional one has been proven to be very effective. Since the experiments chosen are very generic and useful in the Computational Fluid Dynamics field the scope of the `D-adaptive solver` can be even more complex problems with for instance multiple pipes.

Chapter 7

Conclusions. Future Work

In the course of this Thesis, the numerical simulation of single-and multi-phase transient incompressible viscous fluids has been considered. Two important goals have been achieved. The first one was the construction of two efficient solvers for the simulation of single-and multi-phase transient incompressible fluids. The second one was to develop a mechanism for coupling two transient solvers of different space dimension: a one-dimensional solver and a two-dimensional one. This involved two parts: the derivation of new asymptotic expressions for the velocities and volume fractions and the process of error control.

Such coupled solvers open a new methodology for simulating numerical solutions in complex geometries, which has been proved to be efficient, robust and accurate. To the best of our knowledge such methodology is totally new since for the time being there is no references on the literature of such theoretical approach. This has been done on an ad hoc basis. This is the first attempt to include adaptivity in the sense of changing the dimensional variables over the problems domain.

A significant amount of work to develop transient adaptivity multi-phase solvers has been taken. There are several novel aspects, which are outlined below.

Two transient algorithms for single-and multi-phase flows had been developed leading to an efficient solver. The main focus of these algorithms is to solve the time dependent Navier-Stokes equations for viscous incompressible fluids based on two steady solvers developed by Thompson[12] and Lezeau[10]. The discretization strategy, based on finite differences, has been applied for both solvers. For each cell, the velocities were associated with the cell faces and the pressure with the cell centres. In the multi-phase code, the volume fractions were associated with the cell centres as well.

An implicit time dependent algorithm has been chosen with discretization in space, which has been found to be more efficient than an explicit one for the kind of flows that we are dealing with. In these solvers, the discrete equations for both the single-phase and the multi-phase systems have been derived in the finite volume framework and on staggered grids. An important conclusion was the need for an adequate reinitialization process of the initial guess. We mean that after getting convergence to the solution if we do not restart the initial guess, the system will

diverge due the cumulative propagation of the truncation error over the time step iterations. This has been analytically proved for a general case. However the robustness of the algorithm can not be put in doubt because with several initial guesses and for various test problems the solution always converge to the expected steady state solution.

The single-phase transient algorithm, which has been developed from the steady solver `Pamg`[12], emphasises the need of a more powerful algorithm than the original Gauss-Seidel used in the steady solver. This was due to the non-linearity of the Navier-Stokes equations, which degrade the performance of the solver when in the presence of the smooth components of the error. The cost, measured in terms of the number of iterations to get convergence to the solution was extremely high. This difficulty has been overcome using Newton's Method with the support of an automatic differentiation package to compute the Jacobians[18, 19].

Some algorithmic developments have been introduced, due the increased non-linearity of the systems. The use of automatic differentiation for computing the Jacobians involved in the systems has been necessary. In fact, Newton's method implementation for the complex systems that we were dealing has been very difficult due the inconsistency of the residuals and the Jacobians numerically computed. With the automatic differentiation packages[18, 19][20] a high degree of consistency has been achieved, since only the code for the computations of the residuals need to be developed, the computation of the Jacobians was automatic. The use of a novel line-searching to obtain a globally convergent Newton's method has been essential in the multi-phase case. Line-searching relies on the fact that the Newton correction is a descent direction for the Euclidean norm of the residual.

The performance of the two-dimensional single-and multi-phase solvers has been accessed on the following criteria: accuracy, robustness and efficiency.

Solutions for a range of important test problems (channel, bifurcation, joining-T, contraction), of varying complexity have been described, discussed in some detail and compared with solutions provided by independent software. The package chosen has been CFX 4.3. A very good degree of agreement has been obtained between both solutions for all the test cases studied. So we can conclude that the implementation of the solvers is essentially correct and accurate. It has also been proved that the multi-phase solver is able to handle difficult multi-phase flow patterns such as complete phase separation and recirculation zones.

The robustness was established by considering that its convergence is not strongly dependent on the quality of the initial guess for the solution.

Computations on adaptive grids are also supported by the solvers for single-and multi-phase flows. Grids are generated automatically by refining in regions of high truncation error.

A one-dimensional solver `Emaps`, has been used in the course of this project. `Emaps` has been strongly validated in several papers[78, 79].

The first attempt to produce a coupled solver has been coupling the two-

dimensional and one-dimensional solvers described above. Some complex geometries have been used in the test cases, particularly important in the fluid dynamics world.

The interface treatment, when changing from one code to the other, has been carefully developed. Depending on the geometry of each test case, interface regions have been defined. A novel interpolation operator has been developed. Basically only the velocities and volume fractions need to be transferred at each interface, since the pressure fields are computed at each iteration with the updated velocities and volume fractions. New asymptotic solutions of flows in a channel have been derived. These solutions have been validated through the comparison with the computational results for some test cases. Although some limitations on their use which result from the assumptions taken in the derivation process it have been proven to be very useful in the interface treatment. An important point is the technique for deciding when the interfaces should take place. Some discussion has been addressed on this issue. One important conclusion is the strong relation between the velocity on the interface and the distance to geometric features constraints. On the other hand, when changing from a two-dimensional region to a one-dimensional region only the norm of the vertical velocity has to be sufficient close to zero.

The comparison of this first version of `D-adaptive solver` has been done with the two-dimensional transient Pamg for the multi-phase case.

The performance of this `D-adaptive solver` has been measured against the two-dimensional `transient Pamg Multi-phase` regarding the accuracy, robustness and efficiency. Particular attention has been taken with the amount of work involved and CPU-time taken in running the codes. The economy on CPU-time is about ten percent of the corresponding time taken by the two-dimensional code. However some limitations have been verified particularly due to the different coordinate frames used in one-and two-dimensional regions.

Another version of the `D-adaptive solver` prepared to work with cylindrical coordinates has been implemented. This new version has been constructed coupling the one-dimensional solver `Emaps` and the commercial code `CFX 4.3`. Asymptotic expressions for the velocities and volume fractions in cylindrical coordinates have been derived. These have been validated with the computational results of the simulation of multi-phase fluid flows through a pipe. This version of the coupled solver has been validated in a problem with a very complex domain. The validation process has been established by comparing the results produced by this solver with the results obtained by `CFX 4.3`. Focus has been undertaken on the CPU-time taken to achieve the solutions. For the complex test case studied the `D-adaptive solver` has proven to be about thirty percent faster than the commercial code `CFX 4.3`.

One of the main purposes of this Thesis has been to validate the new techniques that can be used in the implementation of D-adaptivity in the coupling of two solvers. These solvers differed fundamentally in the number of dimensions of the variables. Consequently mechanisms had to be undertaken to treat the variation

of the number of dimensions of the variables. The new asymptotic formulas, which have been derived and validated, can be used as a good starting point for other problems where the D-adaptivity can be implemented. For instance the derivation process could be applied to treat the interface of coupling a one-dimensional solver with a three-dimensional one. This reinforces the importance of this contribution to the D-adaptivity field. Another important point is the trigger to use in the process of deciding where to put the dimensional interfaces. During the course of this project several observations of the fluid flow behaviour gave important contributions to the design of the `D-adaptive solver`. Consequently the geometries chosen with the right places to put the interfaces have been based on some rules. These rules rely basically in two main points. One is that the norm of the velocity of the fluid flow has to be lower than the distance to the geometric constraints when changing from a one-dimensional region to a two-dimensional one. Another is that the vertical velocity has to be very close to zero when changing from a two-dimensional region to a one-dimensional one. These rules can be implemented in order to have an automatic `D-adaptive solver`. However this does not mean that we do not have to specify initially where to put the interfaces. Hence the initial guess has to be complemented with this information. Clearly there are a number of pratical developments involved in the automation of D-adaptivity. However, the key issues have been resolved for this class of flow.

This Thesis opens several ways of further research. One interesting task will be the design of a new coarse solver for the `transient Pamg Multi-phase`, which can decrease drastically the computing time, particular for longer transient simulations. A comparison work on the two automatic differentiation packages, used during this project would be useful for latter developments in several fields. Finally the generalisation of the key ideas presented, to other solvers can be extremely important and enlarge the scope of this thesis.

Bibliography

[1] G. K. Batchelor. *An introduction to fluid dynamics.* Cambridge: University Press, 1973.

[2] I. G. Currie. *Fundamental mechanics of fluids.* New-York: McGraw-Hill, 1974.

[3] P. Garabedian. *Partial differential equations.* New York: Wiley, 1964.

[4] K. W. Morton and D. F. Mayers. *Numerical solution of partial differential equations.* Cambridge University Press, 1994.

[5] A. R. Mitchell. *Computational methods in partial differential equations.* John Wiley & Sons, 9999.

[6] Gary A. Sod. *Numerical methods in fluid dynamics:Initial and initial boundary-value problems.* Cambridge University Press, 1985.

[7] R. D. Richtmyer and K. W. Morton. *Difference methods for initial value problems.* New-York: Wiley,2nd ed., 1967.

[8] P. J. Roache. *Computational fluid dynamics.* Albuquerque,N.M.:Hermosa Publishers, 1972.

[9] R. Peyret and T. D. Taylor. *Computational methods for fluid flow.* Berlin,Heidelberg:Springer Verlag, 1983.

[10] Patrick A. Lezeau. *An adaptive quasi-newton coupled multigrid solver for the simulation of steady multiphase flows.* PhD thesis, Cranfield University-School of Mechanical Engineering, 1997.

[11] A. Brandt. Multigrid techniques: 1984 guide - with applications to fluid dynamics. *Computational Fluid Dynamics*, 1984. Lecture Series 1984-04.

[12] C. P. Thompson. Developments in the solution of the incompressible navier-stokes equations: Multigrid algorithms, parallelism and software tools. BSC 91/5, IBM-Bergen Scientific Centre, 1991.

[13] C.P. Thompson, G.K. Leaf, and J. van Rosendale. A dynamically adaptive multigrid algorithm for the incompressible navier-stokes equations-validation and model problems. *Applied Numerical Mathematics*, 9(3):511–532, 1992.

[14] C.P. Thompson, W.R. Cowell, and G.K. Leaf. On the parallelization of an adaptive multigrid algorithm for a class of flow problems. *Parallel Computing*, 18, 1992.

[15] C.P. Thompson. A parallel adaptive multigrid algorithm for the incompressible navier-stokes equations. In *Assymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters (Proceedings of NATO Advanced Workshop on Asymptotic-Induced Numerical Methods for Partial Differential Equations)*, pages 293–309. Kluwer Academic Publishers, 1993.

[16] C. Guardino, C. P. Thompson, C. Omgba-Essama, and L. Hanich. Eulerian multiphase adaptive pipeline solver. Code Specifications Version 2.45, Cranfield University.School of Mechanical Engineering.Applied Mathematics and Computing Group, 2000.

[17] C. Guardino, C. P. Thompson, C. Omgba-Essama, and L. Hanich. Eulerian multiphase adaptive pipeline solver. User Guide Version 2.45, Cranfield University.School of Mechanical Engineering.Applied Mathematics and Computing Group, 2000.

[18] C. Bishop, A. Carle, P. Khademi, and A. Mauer. the adifor 2.0 system for the automatic differentiation of fortran 77 programs. Technical Report Preprint ANL-MCS-P481-1194 and CRPC TR944491, Mathematics and Computer Science Division, Argonne National Laboratory and Center for Research on Parallel Computation, Rice University, 1994.

[19] C. Bishop, A. Carle, P. Khademi, A. Mauer, and P. Hovland. Adifor 2.0 user's guide. ANL MCS-TM-192, Mathematics and Computer Science Division, Argonne National Laboratory, 1995.

[20] J. D. Pryce and J. K. Reid. Ad01, a fortran90 code for automatic differentiation. RAL-TR 1998-057, Rutherford Appleton Laboratory,UK, 1998.

[21] Computational Fluid Dynamic Services, Harwell Laboratory, United Kingdom. *CFX 4.3 flow solver user guide*, October 1995.

[22] N. Curle and H. J. Davies. *Incompressible Flow*, volume 1 of *Modern Fluid Dynamics*. D. Van Nostrand Company Ltd: London, 1968.

[23] D. A. Drew. Mathematical modelling of two-phase flow. In *Annual Review of Fluid Mechanics*, volume 15, pages 261–291. Annual Reviews Inc., 1983.

[24] M. Ishii and G. Kocamystafafaogullari. Two-phase flow models and their limitations. In *Proceedings of NATO Advanced Research Workshop on the Advances in Two-Phase Flow and Heat Transfer*. S. Kakac and M. Ishii, eds., 1982.

[25] H. B. Stewart and B. Wendroff. Two-phase flows: Models and methods. *Journal of Computational Physics*, 56:363–409, 1984.

[26] J. M. Masella, Q. H. Tran, D. Ferre, and C. Pauchon. Transient simulation of two-phase flows in pipes. *International Journal of Multiphase Flow*, 24:739–755, 1998.

[27] I. Faille and E. Heintz. A rough finite volume scheme for modeling two-phase flow in a pipeline. *Computers and Fluids*, 28:213–241, 1999.

[28] D. R. Liles and W. H. Reed. A semi-implicit method for two-phase fluid dynamics. *Journal of Computational Physics*, 26:77, 1978.

[29] C. A. J. Flecther. *Specific Thechniques for Fluid Dynamics*, volume 2 of *Computational Techniques for Fluid Dynamics*. Springer-Verlag, 1st ed., 1988.

[30] Fluent Inc. *Fluent user's guide*, version 5.3 edition, 1995.

[31] S.P. Vanka. Block implicit multigrid solution of navier-stokes equations in primitive variables. *Journal of Computational Physics*, 65(1), 1986.

[32] G. B. Deng, J. Piquet, P. Queutey, and M. Visonneau. New fully coupled solution of the navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 19(7):605–639, 1994.

[33] P. R. McHugh and D. A. Knoll. Fully coupled finite volume solutions of the incompressible navier-stokes and energy equations using an inexact newton method. *International Journal for Numerical Methods in Fluids*, 19(5):439–455, 1994.

[34] V. John and L. Tobiska. Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 33:453–473, 2000.

[35] S. Turek. A comparative study of time-stepping techniques for the incompressible navier-stokes equations: from fully implicit non-linear schemes to semi-implicit projection methods. *International Journal for Numerical Methods in Fluids*, 22:987–1011, 1996.

[36] V. John. A comparison of parallel solvers for the incompressible navier-stokes equations. *Computing and Visualisation in Science*, 4:193–200, 1999.

[37] M. F. Paisley and N. M. Bhatti. Comparison of multigrid methods for neutral and stably stratified flows over two-dimensional obstacles. *Journal of Computational Physics*, 142:581–610, 1998.

[38] D. F. Mayers T. M. Shah and J. S. Rollet. *Analysis and application of a line solver for recirculating flows using multigrid methods.* Numerical Treatment of the Navier-Stokes Equations. W. Hackbusch and R. Rannacher(Vieweg,Braunschweig), 1990. Notes on Numerical Fluid Mechanics.

[39] M. C. Thompson and J. H. Ferziger. An adaptive multigrid technique for the incompressible navier-stokes equations. *Journal of Computational Physics*, 82:94, 1989.

[40] S. Zeng and P. Wesseling. Numerical study of a multigrid method with four smoothing methods for the incompressible navier-stokes equations in general coordinates. In *Sixth Copper Mountain Conference on Multigrid Methods*, page 409. Nasa Conference Publication 3339, 1996.

[41] C. Hirsch. *Numerical computation of internal and external Flows.* John Wiley & Sons, 1990. Computational Methods for Inviscid and Viscous Flows.

[42] M.S. Liou and B.van Leer. Choice of implicit and explicit operators for the upwind differencing method. AIAA 88-0624, American Institute of Aeronautics and Astronautics, 1988. Paper presented at: AIAA 26th Aerospace Sciences Meeting, January 11-14, 1988/Reno, Nevada.

[43] S. F. McCormick, editor. *Multigrid methods.* SIAM. Frontiers in Applied Mathematics 3. Philadelphia, 1987, 1987.

[44] A. Brandt. Multi-level adaptive technique for fast numerical solution to boundary value problems. In H. Cabannes and editors R. Temam, editors, *Proceedings of Third International Conference on Numerical Methods in Fluid Mechanics*, volume 1, pages 82–89. SringerVerlag, 1973. Lecture Notes in Physics 18.

[45] A. Brandt. Multi-level adaptive solutions to boundary value problems. *Mathematics of Computations*, 31:333–390, 1977.

[46] A. Brandt and N. Dinar. Multigrid solutions to elliptic flow problems. *Numerical Methods for Partial Differential Equations*, pages 53–147, 1979.

[47] P. Wesseling. *An introduction to multigrid methods.* John Wiley and Sons, 1992.

[48] William L. Briggs. *A multigrid tutorial.* Society for Industrial and Applied Mathematics, 1987.

[49] R. Varga. *Matrix iterative analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1962.

[50] D. Young. *Iterative solution of large linear systems.* New York: Academic Press, 1971.

[51] A. Brandt and A. A. Lubrecht. Multilevel matrix multiplication and fast solution of integral equations. *Journal of Computational Physics*, 90:348–370. 1990.

[52] S. Murata, N. Satofuka, and T. Kushiyama. Parabolic multi-grid method for incompressible viscous flows using a group explicit relaxation scheme. *Computers and Fluids*, 19:33–41, 1991.

[53] P. He, M. Salcudean, I. S. Gartshore, and P. Nowak. Multigrid calculation of fluid flows in complex 3d geometries using curvilinear grids. *Computers and Fluids*, 25:395–419, 1996.

[54] Hu Hong. Application of an automatic differentiation method to a 2d navier-stokes cfd code. *Computer Methods in Applied Mechanics and Engineering*, 156:179–183, 1998.

[55] J. Steelant, E. Dick, and S. Pattijn. Multigrid methods for compressible navier-stokes equations in low-speed flows. *Journal of Computational and Applied Mathematics*, 82:379–388, 1997.

[56] Jun Zhang. Residual scaling techniques in multigrid, ii: Pratical applications. *Applied Mathematics and Computation*, 90:229–252, 1998.

[57] K. S. Kang and D. Y. Kwak. Convergence estimates for multigrid algorithms. *Computers and Mathematics with Applications*, 34:15–22, 1997.

[58] K. Stuben and U. Trottenberg. *Multigrid methods: fundamental algorithms, model problem analysis and applications.* in Hackbusch and Trottenberg, 1982.

[59] J. Tinsley Oden. Adaptivity and smart algorithms for fluid-structure interaction. AIAA 90-0577, American Institute of Aeronautics and Astronautics, 1990. Paper presented at: AIAA 28th Aerospace Sciences Meeting, January 8-11, 1990/Reno, Nevada.

[60] D. Bai and A. Brandt. Local mesh refinement multilevel techniques. *SIAM Journal for Scientific and Statistical Computing*, 8:109–134, 1987.

[61] M. J. Berger. On conservation at grid interfaces. *SIAM Journal on Numerical Analysis*, 24:967–984, 1987.

[62] D. J. Mavriplis. Unstructured mesh generation and adaptivity. NASA/CR 195069, National Aeronautics and Space Administration, 1995. Lecture notes prepared for 26th Computational Fluid Dynamics Lecture Series Program of the von Karman Institute(VKI) for Fluid Dynamics, Rhode-Saint-Genese, Belgium.

[63] D. A. Mayne, A. S. Usmanu, and M. Crapper. h-adaptive finite element solution of high rayleigh number thermally driven cavity problem. *International Journal of Numerical Methods for Heat and Fluid Flow*, 10(6):598–615, 2000.

[64] D. Pelletier. Adaptive finite element computations of complex flows. *International Journal for Numerical Methods in Fluids*, 31(1):189–202, 1999.

[65] J. Wu, H. Ritzdorf, K. Oosterlee, B. Steckel, and A. Schuller. Adaptive parallel multigrid solution of 2d incompressible navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 24(6):875–892, 1997.

[66] E. Turgeon, D. Pelletier, and L. Ignat. Effects of adaptivity on various finite element sschemes for turbulent heat transfer and flow predictions. AIAA 98-0853, American Institute of Aeronautics and Astronautics, 1998. Paper presented at: AIAA 36th Aerospace Sciences Meeting and Exhibit, January 12-15, 1998/Reno, Nevada.

[67] R. E. Ewing, R. D. Lazarov, and A. T. Vassilev. Finite difference scheme for parabolic problems on composite grid with refinement in space and time. *SIAM Journal on Numerical Analysis*, 31:1605–1622, 1994.

[68] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53:484–512, 1984.

[69] D. B. Spalding. A novel finite difference formulation for differential expression involving both first and second derivatives. *International Journal for Numerical Methods in Enginnering*, 4:551–559, 1972.

[70] B. P. Leonard and J. E. Drummond. Why you should not use hybrid, power-law or related exponential schemes for convective modelling-there are much better alternatives. *International Journal for Numerical Methods in Fluids*, 20:421–441, 1995.

[71] J.M. Ortega. *Iterative solution of non-linear equations in several variables.* New-York:Academic Press, 1970.

[72] C. W. Campbell. Convergence of newton's method for a single real equation. NASA TP 2489, National Aeronautics and Space Administration, 1985.

[73] C. Bischof and W. T. Jones. Experiences with the application of the adic automatic differentiation tool to the cscmdo 3-d volume grid generation code. AIAA 96-0716, American Institute of Aeronautics and Astronautics, 1996. Paper presented at: AIAA 34th Aerospace Sciences Meeting and Exhibit, January 15-18, 1996/Reno, Nevada.

[74] W. H. Press, B. P. Flannery, S. A. Teulolsky, and W. T. Vetterling. *Numerical Recipes in C: the Art of Scientific Programming.* Cambridge University Press, 2nd ed. edition, 1992.

[75] P. D. Orkwis. A newton's method solver for the navier-stokes equations. AIAA 90-1524, American Institute of Aeronautics and Astronautics, 1990. Paper presented at: AIAA 21st Fluid Dynamics, Plasma Dynamics and Lasers Conference, June 18-20, 1990/Seattle, Washington.

[76] Z. Gao. *The numerical treatment of curved boundary surfaces in an unfitted grid formulation for computaional fluid dynamics.* PhD thesis, Cranfield University-School of Mechanical Engineering, 1999.

[77] I. E. Idelchik. *Handbook of Hydraulic Resistance.* Hemisphere Publishing Corporation, 2nd ed., revised and augmented edition, 1986.

[78] L. Hanich and C. Thompson. Validation of a novel alghorithm for the adaptive calculation of transient stratified flow of gas, oil and water in pipelines. *International Journal for Numerical Methods in Engineering*, 51(5):579–607, 2001.

[79] L. Hanich and C. Thompson. Numerical simulation of multiphase flow speed, error control and robustness final summary report. TMF 1:Project 5, Applied Mathematics and Computing Group, Cranfield University, 2000.

[80] T. Myers. Private communication.

[81] A. Georgescu. *Asymptotic Treatment of Differential Equations*. Chapman and Hall, 1995.